Your complete Web design kit: in-depth tutorials **PLUS** CD packed with software

FORMERLY arts specialist

COMPUTER arts projects

SELL YOUR CREATIONS ONLINE

WIN WEB HOSTING WORTH OVER £1000

WEBSITE IN A WEEKEND

Build a money-making site in just two days – we show you how p17

IN ASSOCIATION WITH 1&1

USE PHOTOSHOP TO CREATE STUNNING WEB DESIGNS p18

DESIGN THE PERFECT E-COMMERCE SITE

- Build effective templates in Dreamweaver
- Add a comprehensive search engine
- Incorporate PHP for advanced effects
- Quickly create an online store
- Publish your site to the Web

UK edition · Printed in the UK · Outside the UK and Eire £6

future publishing Media with passion

ISSUE 48 2003 £6

9 771740 900004 08

COMPUTER arts projects

AT A GLANCE

TUTORIALS
18 Website structure and design
26 Creating HTML pages
34 Working with templates
44 Adding a search engine
54 Setting up a shopping cart
62 Publishing your site

FEATURES
10 Getting top results quickly
68 30 tips for faster Web design
72 Building a PHP shopping cart
78 Designers' own online shops

PROFILES
84 Special Moves
88 Skive Creative
90 Tonic Design

COMPETITION
53 Win Web hosting worth £1000

SUBSCRIBE
67 Special subscription offer

REGULARS
07 CD contents
91 Back issues
94 On the CD...
96 Book reviews
98 Next issue

FEATURES

Tips, advice and in-depth articles from Web design experts

10 Speed demons

How do professional Web designers turn out great results for major clients quickly? We reveal their working methods

68 Faster Web design

We've got 30 essential tips to help you work quicker in *Fireworks*, *Dreamweaver* and *GoLive*. Don't get left behind

72 E-commerce for free

A comprehensive guide to building a shopping cart with PHP. And a free solution for making bigger e-commerce sites

78 Market values

Web designers build their own online shops too. We look at 12, including those of Airside, Identikal, Kerb and BD4D

CONTENTS

PROFILES

Find out how the experts turn out great Websites

84 Special Moves

This London agency focuses on creating entertaining interactive sites, like The Osbournes, rather than straight site design

88 Skive Creative

Discover how the Skive team has created a small niche for itself in the ever-growing online games market

90 Tonic Design

These Camden-based Web designers feed heavily off each other's ideas, especially when working to tight deadlines

Firetrap was one of Skive's first clients, and the agency is still proud of this site

WIN! Free Web hosting from 1&1 worth over £1000 See p53

arts projects
WEBSITE IN A WEEKEND
Build a money-making site in just two days – we show you how
USE PHOTOSHOP TO CREATE STUNNING WEB DESIGNS
DESIGN THE PERFECT E-COMMERCE SITE

FULL PROJECT

Follow our 46-page guide to getting your own Website up and running in just two days

CREATE A WEBSITE IN A WEEKEND

SECTION ONE

18 Structure and design

Make sure you know what you're trying to achieve, before defining your templates and doing the initial design in *Photoshop*

SECTION TWO

26 Creating the pages

Use *Photoshop's* Save for Web tool to convert your roughs into HTML pages within *Dreamweaver*, then format the text

SECTION THREE

34 Working with templates

Templates enable you to create Web pages quickly and easily, and help streamline your site in anticipation of future developments

SECTION FOUR

44 Extending the search

We show you how to embed a comprehensive search engine into your site with the minimum of technical knowledge

SECTION FIVE

54 Setting up a shopping cart

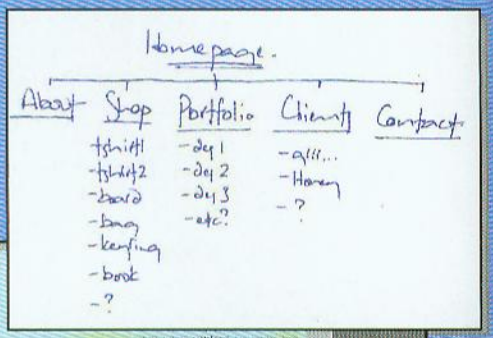
Create a secure online store, complete with pages announcing the confirmation or cancellation of transactions

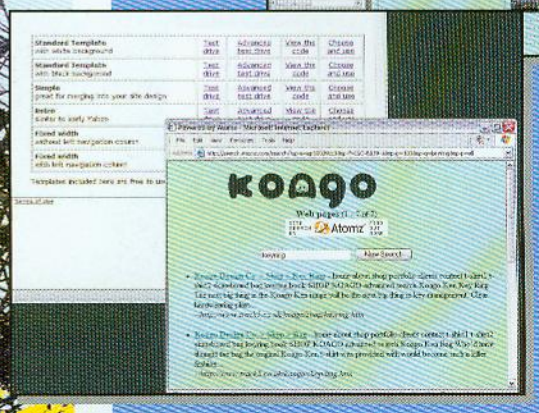
SECTION SIX

62 Publishing your Website

Add meta data to the pages, then check your site and publish it to the Web. Finally, make sure the search engines know you've gone live

TUTORIAL STARTS ON P17





Matthew Harvey's photographic vector images illustrate our Website project

Follow our massive tutorial to find out how to build and publish a money-making Website in a weekend

COMPUTER arts projects INTERACTIVE

SOFTWARE ›

All the programs and extensions you need to build a Website in a weekend...

DUAL DEMO

Dreamweaver MX

Macromedia's HTML editor is the quickest way to put together a professional Website. Using visual layout tools, you can create Web pages with little or no HTML knowledge, while experienced developers will appreciate the code-editing tools on offer. *Dreamweaver MX* combines both visual design and development capabilities in a single tool, which everyone involved in a project can work with.

This trial version of *Dreamweaver* is fully functional for 30 days from the time of installation, giving you ample time to get to grips with it.

PC FULL PRODUCT

EROL 3 Small Store

Traditionally, e-commerce functionality for Websites has been the preserve of big businesses with deep pockets. *EROL 3* is an affordable and flexible application for creating shopping cart facilities. This *Small Store* version enables you to create a store with ten products that can be hosted on any commercial Web space — ideal for those who just want to dabble, or for companies with a limited product range.

EROL includes visual design tools to create your store and has no limitations on store layout. Once your site is online, *EROL* takes care of store management tasks, such as processing orders, creating invoices, sorting orders and keeping track of customer details.

CD CONTENTS

Ten extensions

These *Dreamweaver* extensions enable you to make site navigation, create e-commerce functions and embed rich media in your Web pages. Install *Extension Manager* (also on the CD) to make these available

Dreamweaver templates

Get a head start with five *Dreamweaver* templates courtesy of dreamweaver-templates.net. These pages contain the HTML, images and JavaScript needed to create pages for your online store

Project files

All the raw components required to follow our tutorials are available on the CD. You'll find them in the folders entitled Section 1 to Section 6. The project starts on page 17

COMPUTER arts projects

Subscriptions – page 67
Future Publishing Ltd, FREEPOST, BS4900, Somerton, Somerset, TA11 7B
[t] Customer hotline: 0870 444 8455
[f] 01458 271 146
[e] ca.subs@futurenet.co.uk

Editorial [t] 01225 442 244
Gillian Roach editor
[e] gillian.roach@futurenet.co.uk
Ashley Young acting production editor
[e] caproduction@futurenet.co.uk
Matthew Harvey art editor
[e] matthew.harvey@futurenet.co.uk
Matt Gallimore senior CD editor
[e] matt.gallimore@futurenet.co.uk
Joseph Russ CD editor
[e] joseph.russ@futurenet.co.uk

CD problems [t] 01225 822 743
Technical support: ca.support@futurenet.co.uk

Contributors
Cover: Vår [w] www.woo.se
Illustrations: Halvor Bodin, Matthew Harvey, Stephen Johnston, Richard Llewellyn and Vår
Editorial: Graeme Aymer, Michael Burns, Craig Grannell, Karl Hodge, David Howell, George Mclachlan, Ed Ricketts, Chris Schmidt and Jonathan Wright
Printed at: TPL Printers (UK) Ltd.

Contact
Computer Arts Projects, Future Publishing, 30 Monmouth Street, Bath BA1 2BW [t] 01225 442 244 [f] 01225 732 295 [t] ISDN: 01225 789 293 [w] www.computerarts.co.uk

Advertising [t] 01225 442 244
Rob Wittmaak senior portfolio ad manager
[e] rob.wittmaak@futurenet.co.uk
Bryony Watts senior account executive
[e] bryony.watts@futurenet.co.uk
Andrew Healy senior account executive
[e] andrew.healy@futurenet.co.uk
Lee Rossiter senior account executive
[e] lee.rossiter@futurenet.co.uk
Tony Marshall classified sales executive
[e] tony.marshall@futurenet.co.uk

Production
Katty Pigott production coordinator

Overseas licensing [t] 01225 442 244
[e] simon.wear@futurenet.co.uk 01225 442 244

Management
Steve Jarratt group senior editor
Sheena Pittaway publisher
John Weir publishing director

Distribution
Seymour Distribution, 86 Newman St, London W1T 3EX
[t] 0207 396 8000
Jamie Malley circulation manager
[e] jamie.malley@futurenet.co.uk

The Future Network
Roger Parry non-executive chairman
Greg Ingham chief executive
Colin Morrison ceo and managing director, UK
John Bowman group finance director

All rights reserved. All trademarks and copyrights in this issue are recognised, and are acknowledged where possible. If we have failed to credit your copyright then do please contact us – we'll happily correct any oversight. Any material submitted is accepted on the basis of a worldwide right to publish in printed or electronic form – be sure to clearly label all discs. All contents © Future Publishing Ltd 2003.
Part of The Future Network plc, which has offices in Bath, London, Milan, New York, Paris, San Francisco and Wroclaw.

Future Publishing Ltd is part of The Future Network plc.

The Future Network produces carefully targeted specialist magazines and Websites for groups of people who share a passion. We aim to satisfy their passion by creating titles that offer superb value for money. Today we publish more than 80 magazines and over 30 magazine Websites and networks from offices in five countries. The company also licenses 32 titles resulting in over 60 local editions in 23 countries.
The Future Network plc is a public company quoted on the London Stock Exchange (symbol: FNET) • Non-executive Chairman: Roger Parry • Chief Executive: Greg Ingham • Group Finance Director: John Bowman • Tel: +44 1225 442244 • www.thefuturenetwork.plc.uk • Bath London Los Angeles Milan New York Paris San Francisco Wroclaw

ABC *Computer Arts Projects* is a member of the Audit Bureau of Circulations Jan-Dec 2002: **21,285**

Illustration: Halvor Bodin [w] www.superlow.com

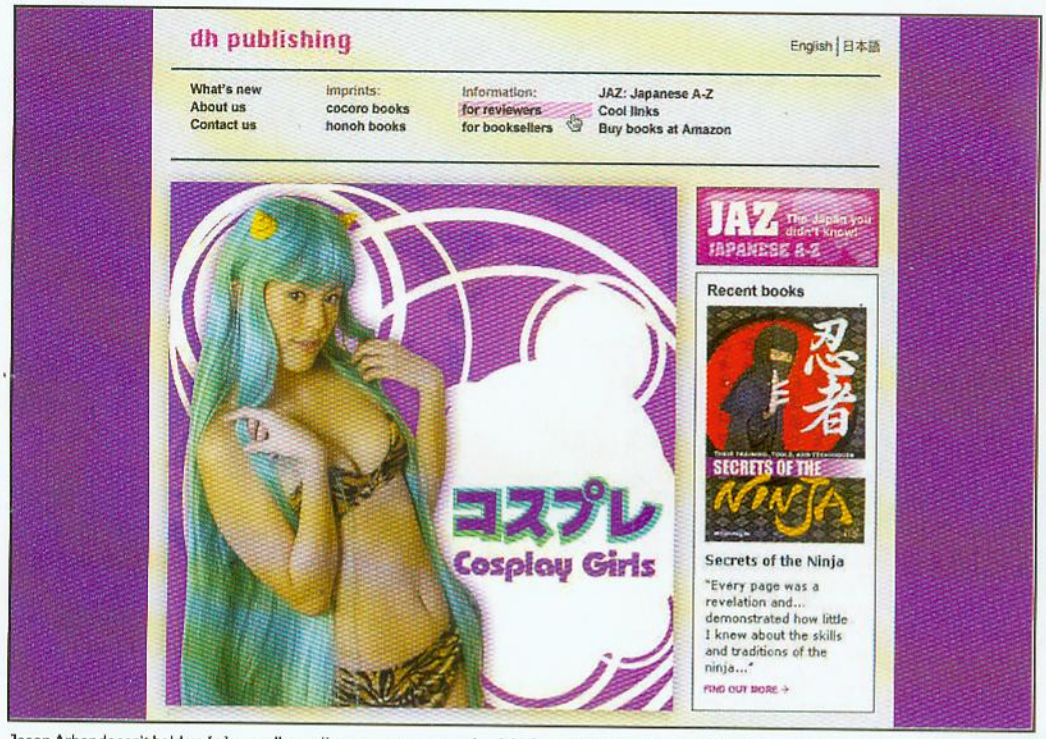
Speed demons

We talk to a number of Web designers about how to work at top speed, and, more importantly, how to get things right...

Every Web designer has been there: a client expresses an interest in using you for "the most important project ever", and then glibly remarks that the deadline was "yesterday". Only they're not joking. In recent times, such occurrences have become commonplace — and taken in tandem with slashed budgets and economic instability, it pays to work fast. We interrogate several designers about how to hone efficiency, from start to finish, when creating a commissioned Website.

Preparation

When time's limited, you might be tempted to get stuck in and immediately start designing, but it pays to plan things through. If you don't, you'll have to make numerous adjustments later — something you'll have scant time for as the



Jason Arber doesn't hold up [w] www.dhp-online.com as an example of the best design he's capable of, but it's a clean, professional-looking site, and a vast improvement on what the company had previously. Not a bad result considering all design, coding and uploading was done within a day.

deadline looms. When deadlines are tight, Mark Johnson of clear* reckons: "It's even more important to sit with a client and establish precisely what their goals are, who the site's audience is, and what information is available. For such projects, we also suggest leaving the architecture as open as possible, to cater for updates later on." Alfredo Silva, Executive Creative Director of Who's We, Inc. cites the importance of gathering finalised material prior to starting a project: "Recovering material after projects begin impedes production at any level. Also, ensure you have constant access to clients, to ask any questions that may arise."

Jakob Nielsen reminds us, though, that customers' needs are also of paramount importance and must be investigated: "By 'customers', I mean the actual users who'll be using the site. Clients rarely know what their customers are really looking to find out, and if a site doesn't answer such questions, it may as well not exist."

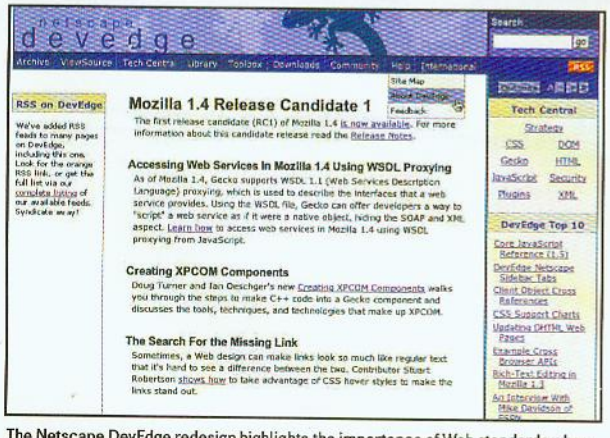
Organisation and design

Once requirements are known and content has been gathered, you need to structure the site. "Go simple," suggests Pixelsurgeon's Jason Arber. "When a deadline's looming, that's not the time to start getting creative with information architecture." Recollective's Chanuki Seresinhe recommends mapping out the entire site in black and white outlines, so content issues can be resolved immediately: "Once the project hits the design stage, the only thing you then need to worry about is look and feel."

Most designers agree that specific software isn't particularly important when creating layouts rapidly. "The most important 'trick' is an understanding of the matter you're working with and a good sense of information hierarchy," claims Kleber's Tom Muller. Alfredo Silva advocates a back-to-basics approach, working on paper to create mock-ups: "We give ourselves short-term milestones regarding layouts, and regroup after a couple of hours to offer suggestions for the product, finally providing a finished sketch to our 3D department."

Whatever medium you favour, it's important that you know it well, as Jason Arber says: "I tend to use *Photoshop*, and have done so since it first came out. I know it inside out, so I can get work done with a minimum of fuss – important when your back's against the wall."

Speeding things up by using stock media seems something of a thorny issue, with many claiming to avoid it entirely, relying on internal 'departments' for photography, 3D work and audio. However, Jason Arber admits that when deadlines are tight: "Photodiscs can be a godsend, although you must make the client aware that such images are not free." He also recommends having a digital camera and scanner handy for creating textures and abstract elements. Tom Muller agrees, and suggests storing such things for later use: "We have libraries of sketches, 3D models, renders and interface ideas that we've developed over time, which can serve us well when working to a tight timescale." Mark Johnson also draws on such resources: "While not something we aim for, if a client presents a really tight deadline, we'll



The Netscape DevEdge redesign highlights the importance of Web standards when having to work quickly – a CSS-driven layout means that cycling through variations is a simple process.

Case study: Netscape DevEdge redesign

CSS guru Eric Meyer recently overhauled Netscape DevEdge, highlighting the value of Web standards when rapidly building a site...

"We reworked the backend, pouring content into a customised XML format, so we could track information about articles. We then decided upon a tables-free design," comments Eric. Separate copies of layout prototypes weren't required – once the example page was created, no one had to touch the HTML again. "All layout work happened in CSS from that point onward," continues Eric. "This meant we could rapidly cycle through variations, getting real-time feedback as CSS was edited on the fly, or by providing alternative themes so people could swap between them."

The design process went more smoothly than expected, largely because browsers have developed rapidly of late. "We created more JavaScript hacks than I would have liked, to work around *Internet Explorer's* limitations," explains Eric. "Otherwise, CSS positioning proved more reliable than we'd previously thought."

The result was a far more accessible site than previously, with smaller pages, all driven by moderately sophisticated CSS that fools some visitors into thinking that tables are still used for layout. ("Not on purpose," says Eric.) The team spent about two weeks on the project – quite an achievement considering its size.

sometimes rebrand a mock-up from elsewhere for quick deployment." This line of thinking is also viable when using *Flash*, as Engage's James Stone testifies: "Over time, you build up a library of modules, including navigation devices, dynamic image viewers and preloaders, and know they work cross-platform. This helps reduce testing time, which often proves to be a hefty part of the process. When generic components are already complete, you can spend your limited time on site elements that'll make it unique."

Building the site

Unsurprisingly, each Web designer has favourite applications when it comes to building Websites. Recollective's Garry Hill sensibly states that when time is at a premium, you should use tools that you're comfortable with: "When under pressure, it's not time to try new ideas that you read about and are keen to try, especially if every time you do something, you have to resort to Google or wade through manuals."

A major point is to ensure that your favoured tool can deal with compliant code. "I always create standards-compliant sites, because hacks and workarounds have a habit of coming back and biting you in the ass," claims Jason Arber – a point that everyone seems to agree on. Using Web standards means sites are more likely to work in the current crop of browsers, and you avoid proprietary tags and features that trip you up further down the line. Of course, how far you go depends upon project parameters, as Eric Meyer states: "If you're building an Intranet site where everyone uses *Netscape Navigator 4.x*, you must keep its abilities (or lack thereof) in mind." Overall, though, he reckons that when using standards, pages end up leaner, and fewer headaches occur when projects are updated.

Modern Web standards help in more than just a technological sense, according to Eric: "Using CSS made my life so much easier – once you're familiar with it, not even point-and-click layout tools are as efficient or flexible." He notes that when creating layouts driven entirely by CSS, HTML rarely needs editing, so rapid prototyping is much easier. Not only that, but editing an existing layout is faster: via CSS, you can easily flip pieces around the screen, meddle with colours and font sizes, and increase or reduce white space. "This can be done with a few keystrokes and a reload in the Web browser," says Eric. "The whole team can see the change and we can automatically check for browser problems as we go."

"Hacks and workarounds have a habit of coming back and biting you in the ass" – Jason Arber

It's easy to get carried away with technology and make basic errors, though, as Jakob Nielsen reminds us: "You must pay special attention to the usability of the site's presence in search engines because that's how many prospects will find it." He explains that this is often done by the most basic of HTML and page content – things often overlooked when working at speed: "Page titles must stand out and make sense when taken out of context and crammed into a listing of ten sites, and query terms that customers might use to find the site should be within the text."

Jakob advocates stopping to check draft designs against checklists of established usability guidelines from prior user research. "Such reports and books, including my own *Homepage Usability*, encapsulate knowledge about what works well with users, and you'll ›

Case study: DH Publishing design

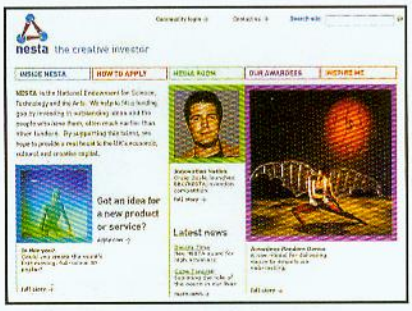
Pixelsurgeon's Jason Arber turned around DH Publishing's site within a single day...

"They liked the original design, which was a great timesaver," says Jason. "There's nothing worse than working to a tight deadline, but a client requiring 50 different layouts."

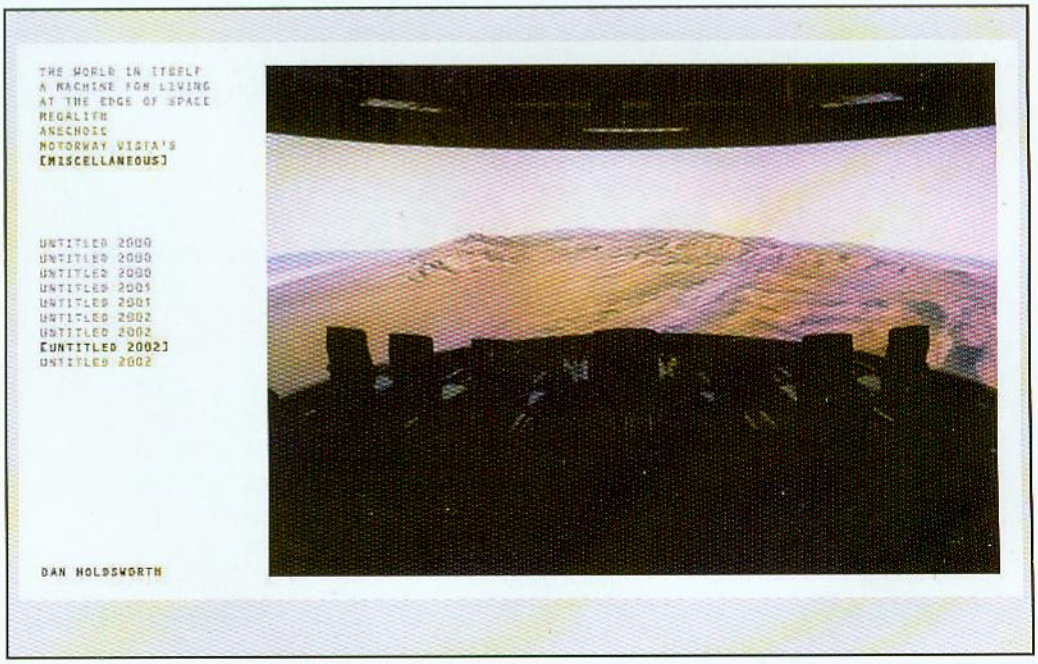
The straightforward layout was intentional, as it enables DH Publishing to edit pages easily and add new content. Unfortunately, the Japanese ISP only allowed the basics of Web technology onto its servers: "I wanted to create a CMS via MySQL and PHP," explains Jason. "But they wouldn't even allow CGI scripts – in the end, I went for straight HTML, JavaScript and a touch of *Flash*."

The site was built quickly because it was effectively a freebie, and Jason had little time to devote to it. He notes, though, that you should try not to forget standard design aspects when in a hurry: "I overlooked using a preloader for the homepage's *Flash* movie – broadband users won't notice, but others will see the weird way the movie loads, and I've not yet found time to fix it..."

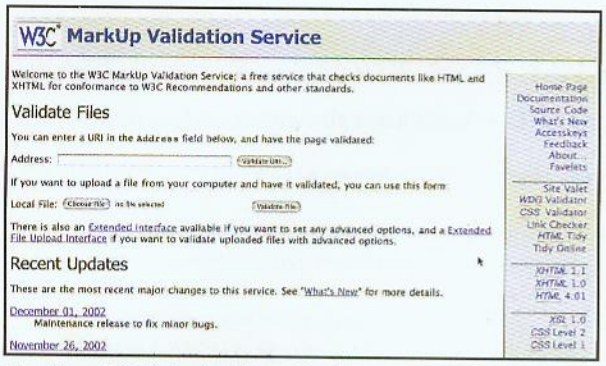
Jason continues: "I'd also have liked to refine the design – I'm sure given two weeks to mull it over instead of a couple of hours, it would have been far better. However, it's a vast improvement on what they had previously, which is the main thing."



Recollective had to turn around [w] www.nesta.org quickly after the organisation rejected a design from another company that didn't meet its needs. Excellent project management and streamlining the building process – designing one section while building others – ensured the project was completed on time.



James Stone's site for photographer Dan Holdsworth – [w] www.danholdsworth.com – was created in two days, because of a small budget. Built in tandem with graphic designers Made Thought, the team crafted a system whereby Dan could update the site himself.


W3C MarkUp Validation Service
Welcome to the W3C MarkUp Validation Service; a free service that checks documents like HTML and XHTML for conformance to W3C Recommendations and other standards.
Validate Files
You can enter a URI in the Address field below, and have the page validated:
Address:
If you want to upload a file from your computer and have it validated, you can use this form:
Local File:
There is also an Extended Interface available if you want to set any advanced options, and a Extended File Upload Interface if you want to validate uploaded files with advanced options.
Recent Updates
These are the most recent major changes to this service. See "What's New" for more details.
December 01, 2002
Maintenance release to fix minor bugs.
November 26, 2002
Home Page
Documentation
Source Code
What's New
Accesskeys
Feedback
About...
Favelets
Site Valet
WDG Validator
CSS Validator
Link Checker
HTML Tidy
Tidy Online
XHTML 1.1
XHTML 1.0
HTML 4.01
XSL 1.0
CSS Level 2
CSS Level 1

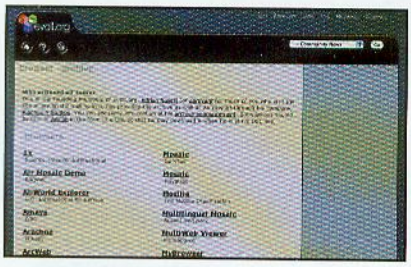
It's not the prettiest of sites, but [w] www.w3.org has many tools to aid Web designers in a hurry. Its validation tools can be a tad picky, but they often point out errors you'd otherwise miss, some of which might cause inconsistencies across browsers.

usually find several 'obvious' things have been overlooked in the rush to do a design quickly."

Taking things further

While it's fair to say that when deadlines are extremely tight few people build anything more than a basic site, there are still alternatives to HTML. As previously mentioned, *Flash's* greater degree of design intricacy is partly offset by reduced compatibility problems, but there are still things to bear in mind when using *Flash* and designing quickly. "It's best to prototype directly in *Flash*," says James Stone. "It's important that the feel of the site works, so you shouldn't rely on static *Photoshop* layouts." He suggests avoiding too many effects and animated elements, because they're often used to try and cover up a 'quickie' design that lacks substance: "Only use animation for a purpose – there's little worse than ploughing through animations, only to arrive at something insignificant." And remember that creating such animations will eat into your limited development time.

With regards to non-*Flash* sites, there was some disagreement between those we spoke to regarding templates. "They can be a great timesaver, but it depends on the size of the site," says Jason Arber. "There's a trade-off between the time it takes to create templates versus the time it takes to create the pages by hand. Of course, if something like the navigation needs redesigning, you'll be burning the midnight oil again changing each page if you don't use a template..." Others, such as Tom Muller, create templates by default, regardless of deadline pressures: "Once a design is signed off, we create templates for each site section, enabling us to manage content via bespoke CMS or XML feeds. Subsequently, we can work fast on the development side, making updates on-the-fly."



Even when a deadline's looming, there's still little excuse for not testing a site in various Web browsers. Many users will be using legacy kit, and while you can't cater for everyone, at least install the last revision or two of each major browser – [w] http://browsers.evolt.org archives many of them.

technologies are those you find licensed as 'open source': "You can sometimes download solutions that work almost immediately, and the general quality of both them and the related documentation is often high." Again, he recommends being totally familiar with the technology when working in this fashion: "Trying to force a bit of software when you have no idea how it works is a mug's game and best avoided."

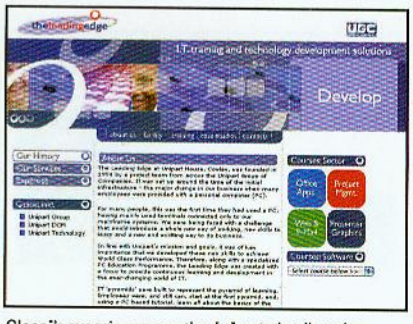
> *"Trying to force a bit of software when you have no idea how it works is a mug's game" – Garry Hill*

Typically, advanced features should be avoided if deadlines are tight. "Shopping carts take time to set up and, because people's money is at stake, you must ensure everything is working correctly," explains Jason Arber. "You can't rush stuff like this, so it's better to explain to a client that they should push back the deadline, or launch the site without the shop element and add it later." If determined to press ahead regardless, Garry Hill suggests the best

Testing your site

Once you have something vaguely resembling a site, you'll have to ensure that it works. Testing can be a lengthy process – something you can't afford when you only have a day or two to complete your project. "In such cases, you should test as you go," suggests Mark Johnson. "That way, you can catch bugs early on, making them easier to fix." As for which browsers you should test in, that's a matter of opinion. "As many as you can get your hands on," suggests Eric Meyer. "You can never do too much testing, and the more cross-browser testing you do, the more you get a feel for what's practical and what isn't."

Another of Who's We Inc's sites – [w] www.handheldgames.com – proves that *Flash* is a viable technology, even if deadlines are tight. The entire project, including design, animation and sound, was constructed in six days, largely due to long hours and strict project management.



Clear "'s experience creating [w] www.leadingedge.uk.net is a warning to all those who underestimate the capability of clients to throw a spanner in the works. Despite sign-off, the client demanded considerable number of changes, causing earlier work to be abandoned. Strong knowledge of both client and technology meant the site still arrived on time.

Who's We, Inc's site for Majesty Records – [w] www.majestyrecords.com – was created in four days. Progress was swift, partly because the client gave the designers creative freedom and also because the site isn't dependent on heavy content.

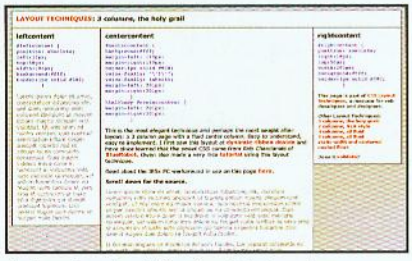
Of course, when time's tight, testing in every permutation on every platform isn't viable. It goes without saying that *Internet Explorer* for Windows should be your first port of call, as the majority of Web users are running it. Don't forget that a substantial amount of people never upgrade their browser, so you should check in at least one previous version, too. If you don't have multiple machines lying around, *VirtualPC* can help, enabling you to run multiple PCs, each with a unique install of *Internet Explorer*, on a single Mac or PC.

However, as Jason Arber states: "The worst thing you can do is make sure your site works with *Internet Explorer* and ignore everything else. Microsoft has a superset of tags I like to call MSHTML, which go beyond W3 standards, with proprietary features." He reckons that when testing in *Internet Explorer* only, it's all too easy to create a site that excludes other browsers and platforms, and suggests running pages through W3's validator at [w] validator.w3.org, which picks up on coding errors and non-standard elements. When you have generic code that's confirmed compliant, store it on your machine for later use.

You should test in a *Gecko*-based browser, too, such as *Mozilla* (in fact, this can make a better starting point, due to its adherence to Web standards) and, if possible, *Safari*, which takes care of KHTML and looks increasingly likely to become the most popular Mac browser. Whether or not you build for version 4 browsers depends on your client's intended market, and can prove time-consuming. When using CSS-driven layouts, you can always 'hide' Style Sheets from such browsers by using @import, which ensures users can see the content, if not the design.

Those creating *Flash* sites have fewer problems, as most browsers are *Flash* friendly. Despite this, don't skimp on the containing HTML pages when designing quickly: remember title tags (don't leave them as Untitled), and ensure any related JavaScript works cross-platform.

Jakob Nielsen also reminds us to test with real users, rather than just checking for problems with the technology and compatibility: "People tend to skip this when in a hurry, but you can do a simple test with several people in only a few hours. There's no need to compose a fancy report on a compressed project, but do run the test and implement as many of the findings as you have time for. Any remaining findings can go on a to-do list for any subsequent redesign."



Having a decent set of bookmarks helps when deadlines are stretched to breaking point. Forums enable you to get rapid responses to problems, while the likes of [w] www.glish.com/css provide freely available starting points for specific areas of Website design and development.

And finally

Ultimately, planning and common sense are your closest allies when deadlines are tight. Things should always be relevant: if a site doesn't require *Flash*, don't convince a client to employ it. Instead, use tools to create something that's functional, interactive and visually pleasing, within your set timescale. Tom Muller reminds us to bear in mind things common to any successful Web design project: "Plan ahead, build a site with solid architecture, and if possible, implement a CMS tool that manages content and graphics. Also, name and file your assets sensibly and logically." James Stone advises to always have a clear goal of client requirements and to be realistic: "When working at speed, you must work hard *and* smart. Don't promise what you can't deliver – if a client is asking for the Earth, let them know." And finally, one all-important, technology-free tip from Pixelsurgeon's Jason Arber: "Don't skimp on the caffeine the night before the deadline..." CR•D

Expertise provided by Craig Grannell. [w] www.snubcommunications.com.

Dos and Don'ts

Even when working at speed, there are certain rules that all good Website designers should adhere to...

Do...

- ...compile stock material as you work on projects. Keep handy reusable bits of code, mark-up and scripts, along with textures and other stock media.
- ...know your software and underlying technology – learn keyboard shortcuts and built-in time-saving devices.
- ...utilise Web standards. Valid code means sites will more likely work across all Web browsers, reducing bug fixing, while CSS layouts enable rapid prototyping and variations.
- ...make use of popular Web design forums. If you can't fix a problem, you can often get rapid, helpful feedback in minutes, at any time of the day.
- ...plan thoroughly, even within restricted timeframes. Ensure you have all finalised material prior to commencing work, and get sign-off from the client at each milestone, to minimise alterations later on.

Don't...

- ...overcomplicate things, such as information architecture and graphic design. When deadlines are looming, that's not the time to start getting flamboyantly creative.
- ...just create sites for *Internet Explorer* in Windows, as this will come back to haunt you when the client complains that their customers using alternate browsers and platforms cannot access the site.
- ...be tempted to lift someone else's design and ideas. At best, you'll be shamed by the design community; at worst, you could find a lawsuit winging its way to you.
- ...promise the earth – you must be realistic when deadlines are tight, so don't oversell. Ensure clients are aware of what's possible within given timeframes. If necessary, phase a site's development – don't go for the big launch in the wrong timeframe and miss both!
- ...ignore the client in the hope that your ideas will all be approved automatically, and don't waste precious time on concepts without a solid idea of what's required.

Illustration: Matthew Harvey
Photography: Richard Llewellyn

SECTION **ONE**

Structure and design

Start by understanding the brief, building the site structure and planning the design...

Clients huh? Don't you just love 'em... It's Friday afternoon and somehow you've been landed with a site that needs to be built for a product launch on Monday morning.

The client has provided a (brief) brief outlining the content and creative preferences for their site. They've emphasised that it'll be predominantly based around a searchable shopping cart making their new products available to the world when the advertising campaign kicks in on Monday morning. They also want a bit of background information on the company and their clients, but the goods must be up for sale on Monday...

There isn't time to mess around with bespoke systems if everything is to be completed over the weekend, so we're going to make use of existing *Dreamweaver* extensions to get a functional site up and running. But as the client has promised that more time (and money) will be available after the product launch to refine the Website into something a little more professional, we'll be anticipating features within the layout and design that may benefit from future development. These might include a more universal shopping cart, a site-specific search facility and the introduction of a database to manage content more efficiently.

But for now, the clock is ticking and there isn't a moment to waste on dragging techies into the fray. The time limitations mean that it's all down to you to get it done over the weekend.

That's the scenario for our project and, in the real world, it's not as ridiculous as you might think. There are many features in our Koago site that can be integrated into your own Website. As a designer, you may want to use some of the principles here to create your own personal image library offering your work for sale online. In that case, both the shopping cart and search facility will be invaluable.

But all that's just going to have to wait until after the weekend... >

Expertise provided by Chris Schmidt [e] chris@track5.co.uk. Chris is a freelance writer who regularly contributes to various magazines, including *Computer Arts Projects*, *Computer Arts*, *.Net* and *PC Plus*. After almost ten years of Web-production experience working for numerous design agencies and too many clients to keep track of, he also builds sites for his own company, track5.co.uk.

The site brief and content can be found on the CD along with all assets for this project. Visit the final site at [w] www.koago.com to see how the search and cart features operate.

Part 1: Understanding the project

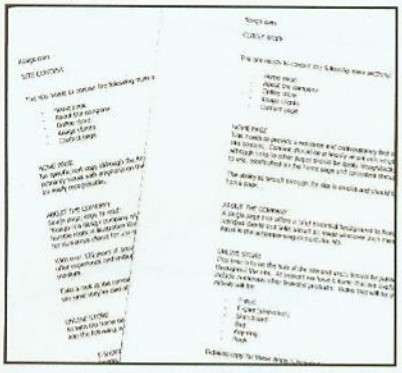
Before starting the design, you need some background to the project. Don't do anything without a brief...

Planning

As with any project, planning is a crucial stage that shouldn't be overlooked. Use a client's brief as the groundrules and expand if necessary to include your more experienced ideas. However, always make sure the client understands and approves any ideas before you continue with the development of the site.

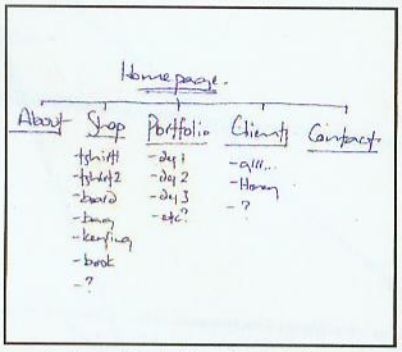
Theme and variations

Plan as many alternative layouts and come up with as many design ideas as you can during the initial stages of your site. For any number of reasons, things will inevitably change or have to be adapted. But having drawn up multiple ideas, you'll be able to anticipate variations or include alternative ideas as the project evolves.

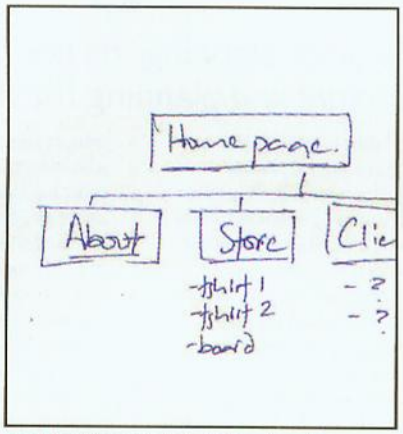


1 Open the brief.doc and content.doc files from the cover CD. These were provided by the client as the blueprint of the site. During initial meetings you can't emphasise enough to the client the importance of a brief – ask for design preferences as well as anticipated content and functionality.

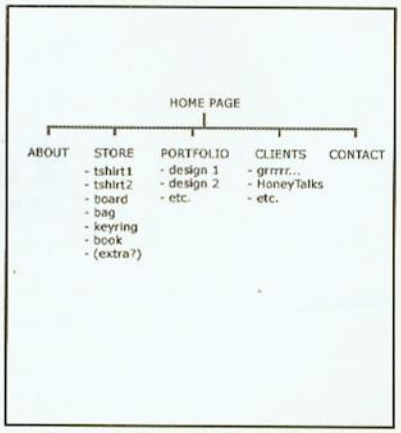
2 The text content will undoubtedly provide a natural structure for the site, so group common areas such as company information, its portfolio, merchandise and so on into logical sections to start planning the site's basic hierarchy.



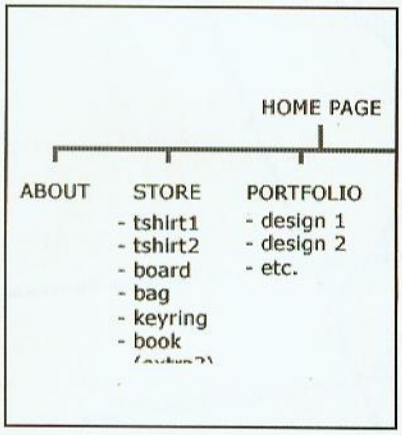
3 Go through the brief again to include specific pages for these sections, while trying to anticipate areas that may be expanded in the future. By its very nature, a Website needs to be adaptable to new and changing content. Any future-proofing you can put in place now will make the site easier to manage later.



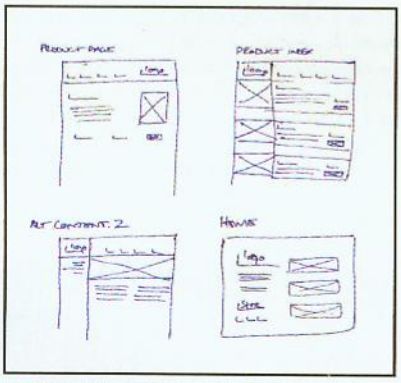
4 Don't forget to include the various home or index pages your site will rely on. A client may have outlined a number of products in their brief, but you'll need to present these in a central page from which users can get an overview and access the individual items.



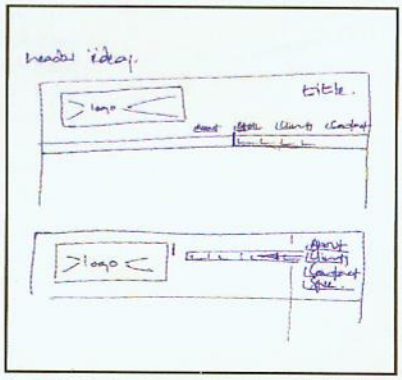
5 Once you're happy with the structure, it's worth recreating it in a clear diagram that you can print out and pin next to your monitor. This useful reference point will ensure that the structure remains constant throughout the development of your site.



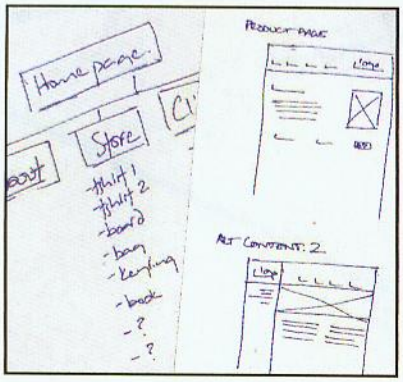
6 It'll also lend some suggestion as to how the site navigation might work best. The most obvious method may not be the most original but it'll probably be among the most intuitive, which is an important feature of any site – particularly one that has a shopping area.



7 The content will also provide some preliminary ideas for page layout and weighting as you determine what needs including. Mock up some initial layouts on paper to give yourself a few options in anticipation of the design stages.



8 By mocking up some layouts, you'll start to recognise which pages or areas will share common designs. You'll also be able to consider how the pages might be constructed to include consistent design features throughout.



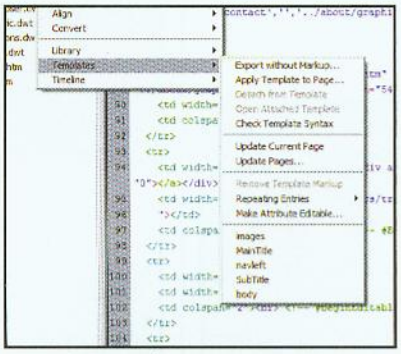
9 The mock layouts also provide guidance as to which pages can be grouped together using common *Dreamweaver* templates, which will help to speed up the process of constructing the site as well as making global updates much more efficient.

Part 2: Defining templates

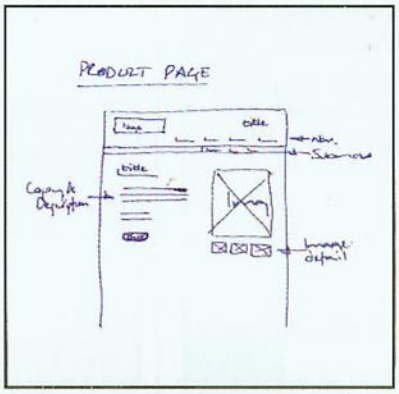
Speed up the eventual production workflow by planning the required templates...

In perspective

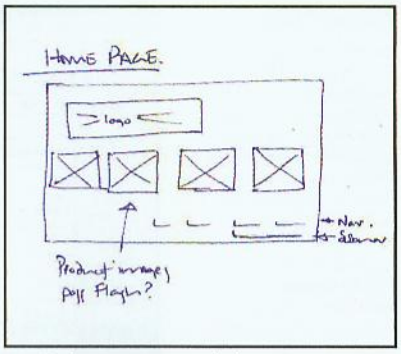
At this early stage, it's a good idea to consider any templates your site may come to rely on later. This will reduce potential problems your site may encounter and will lead to a more efficient workflow as the production of your site develops. Understanding how *Dreamweaver* templates work is a crucial factor if you anticipate their use, but we'll look at them in more detail in Section Three (starting on page 34).



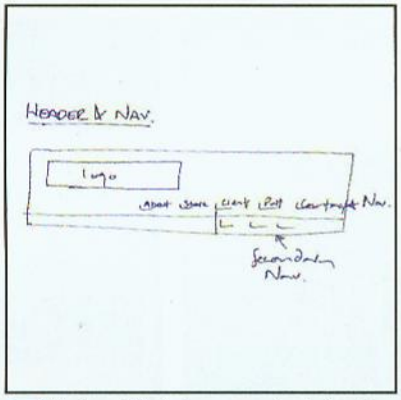
1 Making use of *Dreamweaver* templates enables you to create a few common pages that can be associated with certain sections of the site. We'll look at them in more detail when we get round to creating HTML pages, but for now you just need to determine which pages can be controlled by which templates.



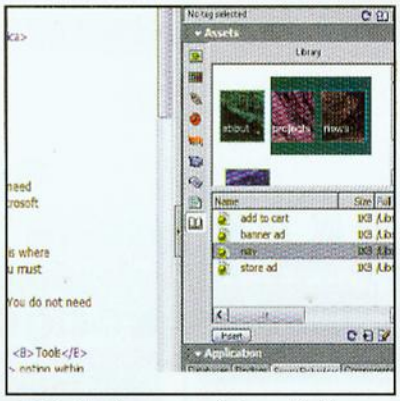
2 The page mock-ups we created previously lend themselves to natural suggestion for possible templates. You'll probably have found that you've ended up considering generic content and product pages as well as a unique homepage.



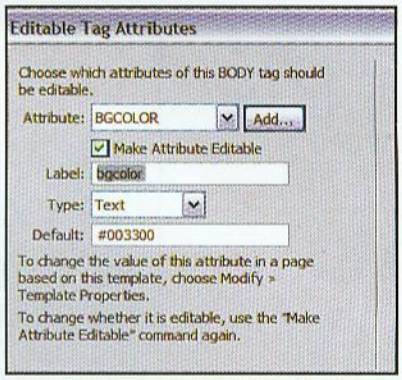
3 Such unique pages as the homepage won't benefit from being used in the context of templates. The main idea behind templates is that common elements, such as navigation, can be updated from a single master template that controls multiple pages; unique pages are obviously independent of these features.



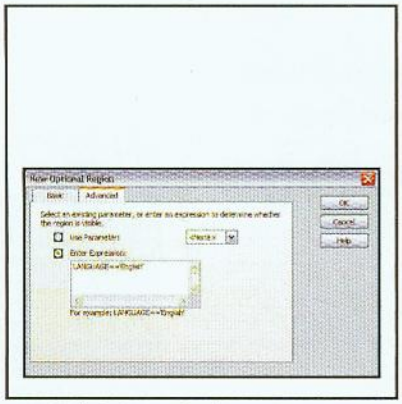
4 Having said that, there are methods of nesting templates. (We'll look at these later in Section Three.) Therefore, even at this early stage, it's worth identifying elements of a page that may be common by design, but independent by function from the parent page (such as navigation).



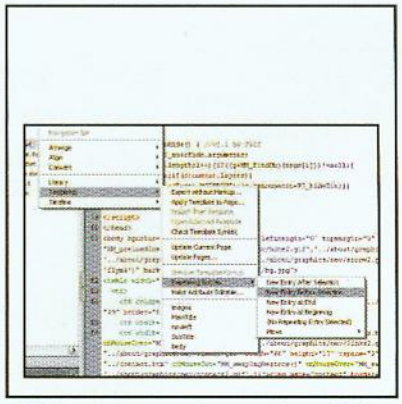
5 Another *Dreamweaver* feature related to templates that's worth keeping in mind is Library items. While templates allow entire pages to be reused and managed, Library items are intended for individual design elements such as navigation – these can be found among your site assets.


Editable Tag Attributes
Choose which attributes of this BODY tag should be editable.
Attribute: BGCOLOR
Add...
Make Attribute Editable
Label: bgcolor
Type: Text
Default: #003300
To change the value of this attribute in a page based on this template, choose Modify » Template Properties.
To change whether it is editable, use the "Make Attribute Editable" command again.

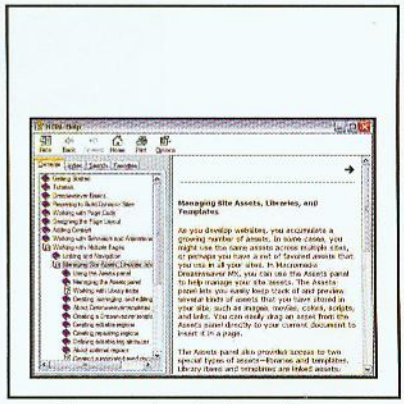
6 Other advanced features of templates that you should be aware of include Editable Tag Attributes. These enable you to define elements of a tag itself that can be modified in subsequent pages, allowing for seemingly minor changes to be accepted without requiring the creation of entirely new templates.



7 Similarly, Optional and Repeating Regions are useful for pages containing varying product lists. This is because they're treated as variables that are displayed or used depending on their defined values. This provides control without having to resort to overly complex nested table options.



8 These features are all accessed along with the majority of Template controls via the Modify>Templates option, which lists both common and more advanced feature sets. Don't forget to consider the Insert bar's Template options for more common functions.



9 This process of constantly amending your initial ideas in consideration of potential production implications will help with planning the design and layout, and will ease the creation process of the site. Don't be afraid of using *Dreamweaver* help files to learn more about these features.

Part 3: Designing the homepage

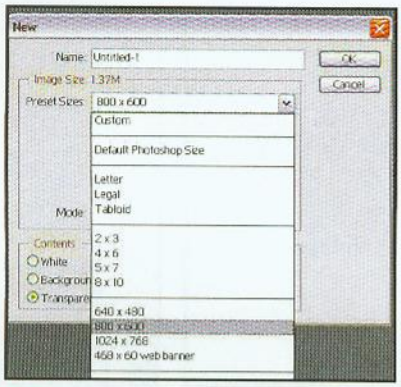
Take your initial ideas into *Photoshop* to develop the homepage design and layout...

Browser sizing

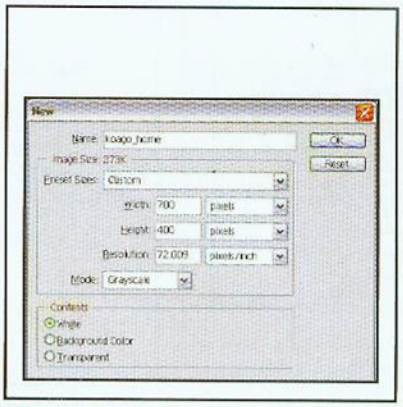
Deciding on the right size for your page layout is tricky because the best render of your site through the browser depends on the end user's screen monitor size and screen resolution. Make it too big and you'll force sideways scrolling, which isn't recommended. Consider the 800x600 resolution to be a reasonable standard and plan the best layout within this letterbox, making sure to compensate for the browser interface, scrollbars and so on.

Grids or guides?

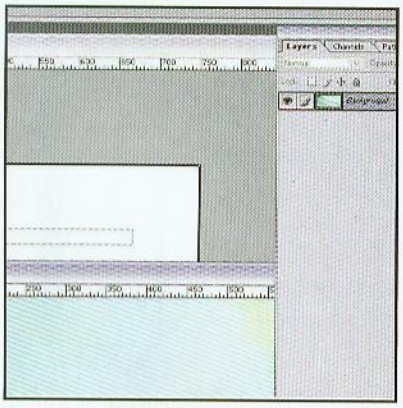
While some people prefer to work with their own positioned guides dragged from the ruler, others prefer to use grids or even a combination of the two. There's no right or wrong method as both provide a similar function. Toggle them into action via the View menu and make use of the Snap To options if you want objects to align themselves automatically when placed nearby.



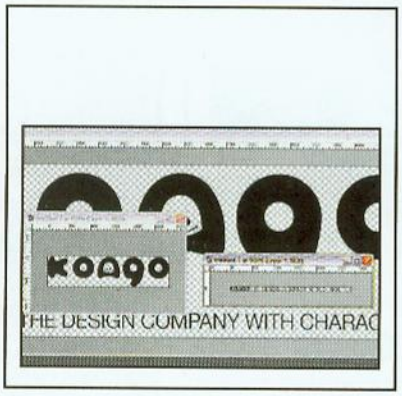
1 Launch your preferred image editor and create a new file – we're using *Photoshop*, but any equivalent application will do the trick. You could use the Preset Sizes drop-down to select a suitable screen resolution, but see the 'Browser sizing' margin note for reasons against this.



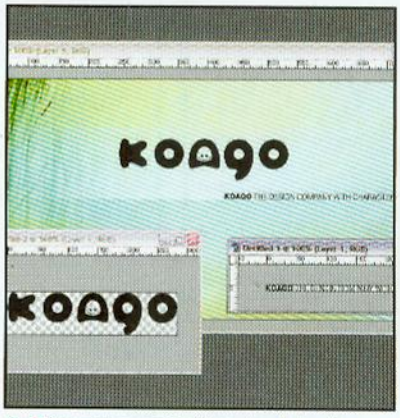
2 Instead, manually configure this to be 750x400 pixels using a resolution of 72dpi to match the screen. This gives you the proportions to work on the homepage which, as a unique page, allows you a little creative freedom to work on your layout.



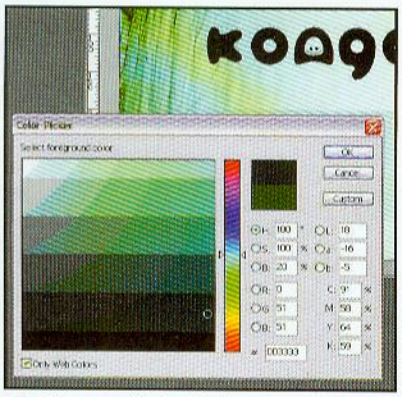
3 As Koago is a design company, it's given us free rein to use its image library. Open the background.jpg file and drag from the Layers palette onto your new canvas while holding down the Shift key to centre the image into place automatically.



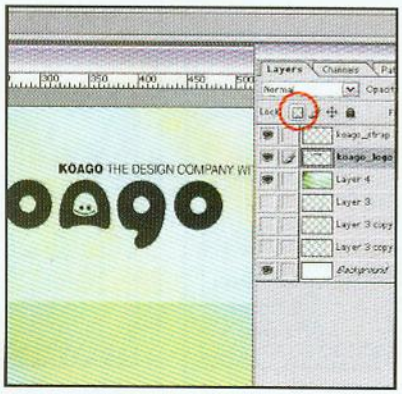
4 Next, open the koagologo.jpg file and convert the background layer to a regular one. Click the white background using the Magic Wand tool and choose Select>Similar. Hit the delete key to isolate the copy, scale down to 25 per cent and split the logo and strap into two separate files.



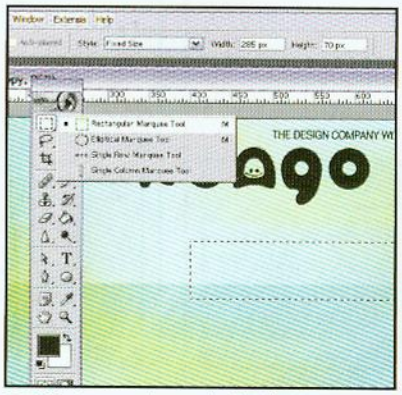
5 Return to our original layout and drag the amended logo and strap graphics onto the canvas to create new layers for each automatically. As the number of layers will increase, it's advisable to get into the habit of naming layers.



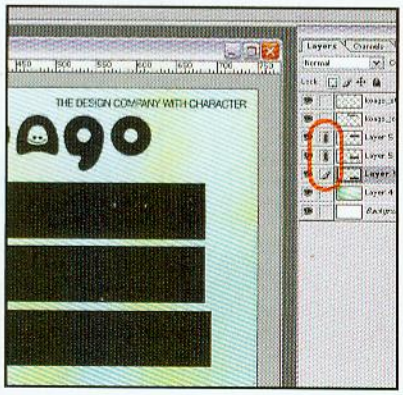
6 The current black of the logo is probably too intense for our design, so take a dark sample from the background image. Go into the Colour Picker and ensure the Only Web Colours option is checked to snap into place – we've settled with #003333 as our colour.



7 To apply this to the logo and strap layers, activate each layer's Lock Transparency option and simply fill the layers in turn. This provides a quick and clean colour replacement without you having to worry about any rogue anti-aliasing that may go amiss.



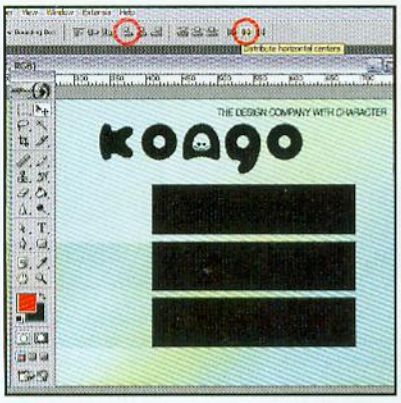
8 Create a new layer and select the Marquee tool. Choose the Fixed Size style from the tools option bar and fix this at 285x70 pixels. With the new layer selected, click on the canvas to generate a selection box at the specified size and fill with a contrasting colour.



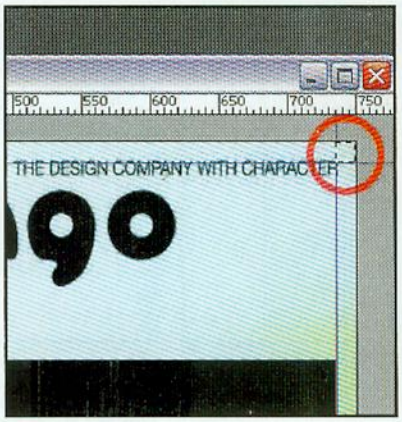
9 Create two duplicates of this layer (Ctrl-J or Command-J) and place them vertically to the right of the design. Don't worry about being too precise about their positioning as you can perfect that in a moment – with them roughly in place, link the three layers together in the Layers palette.

Link states

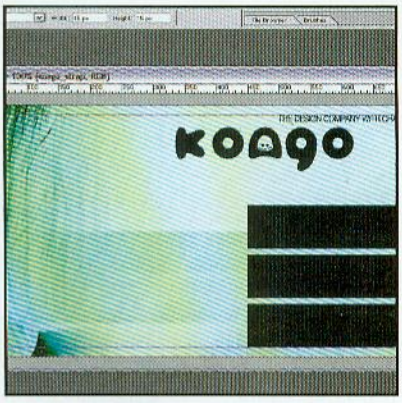
The navigation makes use of CSS controls over the HTML to indicate the link's status. There are a number of states a link can be in, but the two main considerations are for Normal, when there is no user interaction, and Over, when the mouse is passed over the link. Using CSS, you can draw attention to links as we do here by highlighting a link when the user passes over them.



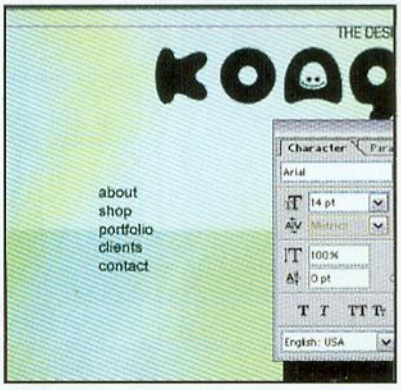
10 Next, choose the Move tool and you'll notice that a range of alignment options become available in the related options bar under the menu. Use the Align Left Edges and Distribute Vertical Centres options for pixel-perfect alignment of your image placeholders.



11 To align these within the page, drag guides around the canvas 15 pixels from the edge to create a common border. Fix the size of the Marquee took at 15x15 pixels, placing an instance in a corner point and snapping guides into place.

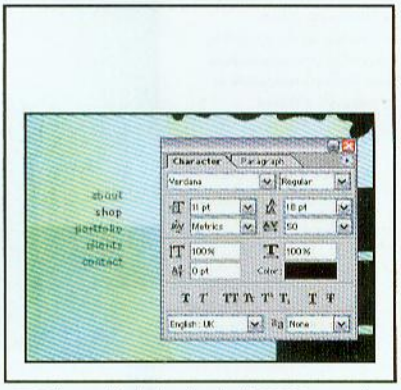


12 Repeat in the diagonally opposite corner to complete the border guides. Drag the strap layer and the locked image placeholders into place at the top and bottom of the layout. As well as ensuring perfect placement, using guides in this way helps to anticipate the construction of the HTML page.

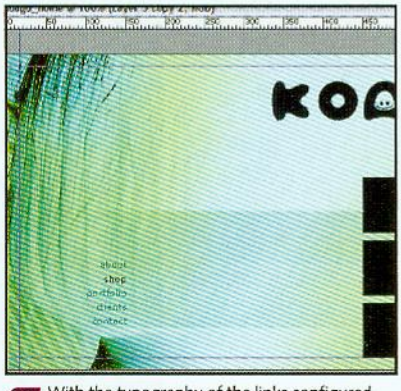


13 We're going to keep the navigation clean and simple, ultimately using regular HTML links enhanced with Cascading Style Sheets (CSS). Select the Text tool and click on the canvas. Type the above options into a single layer using the return key to wrap onto the next line.

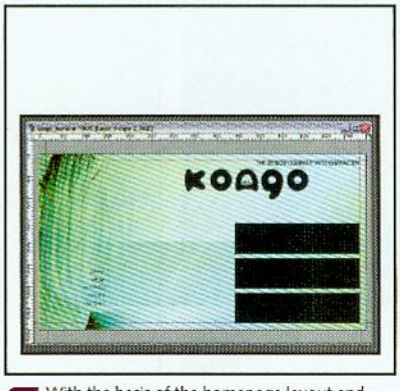
14 Change the font character to the universal Web font Verdana at 11pt. Increase the leading to 18pt, the kerning to 50 and remove any anti-aliasing to emulate HTML text as it might appear in the browser. Move onto the Paragraph options and align the text to the right.



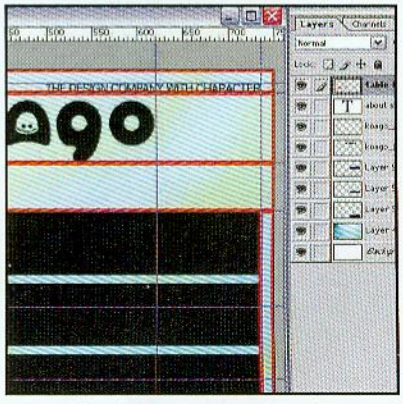
15 Change the text colour to #009999, a couple of Web-safe shades lighter than the logo colour, and then select a single word and revert that to the original #003333. This gives some visual indication of the normal and rollover states for the CSS link – see the margin note for more on this.



16 With the typography of the links configured, you can now move the navigation into place in your design. You might choose to enhance the way this functions to include text descriptions when a rollover is detected, but as time is short for this project we'll put this on hold...



17 With the basis of the homepage layout and design established, now is a good time to review the brief and make sure we've included everything that was requested. We've considered a few features that we might want to develop, but now's the time to make any last minute changes.



18 Finally, take a moment to create guides around your layout to suggest how the HTML table might be constructed. Then, when it comes round to building the page, you'll be able to toggle these into view and recreate the layout efficiently.

Part 4: Creating a contents page

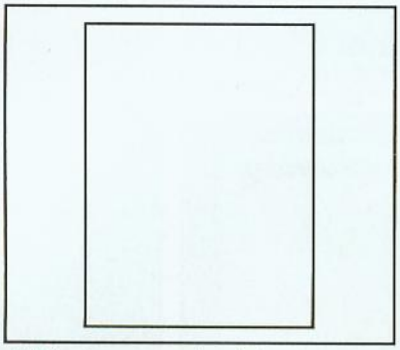
Develop the regular page design considering content and *Dreamweaver* templates...

Design order

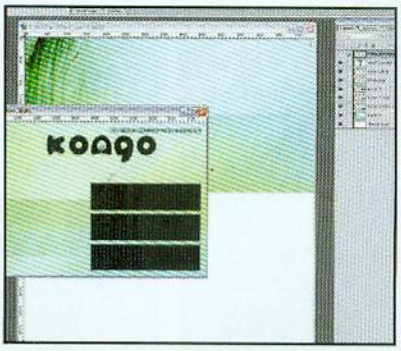
Deciding whether to mock up the homepage before the contents page is a personal choice – you may prefer to do things the other way round as there's no right or wrong way for this. Either way, you'll probably find that elements of your design evolve from earlier ideas, and it's often worth revisiting earlier designs and amending details that may add consistency between the two versions.

Web-safe colours

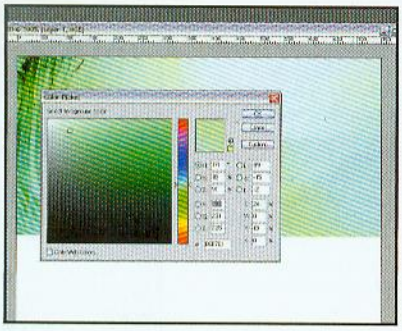
As a rule, it's generally best to keep to the Web-safe colour palette, although with the improvement of computer graphics cards you can ignore this rule on occasions. Because our site is for a design company, we've made the assumption that most site visitors will be able to render non-safe colours successfully, but this is a decision that must be considered carefully because the site may be difficult to view on some systems.



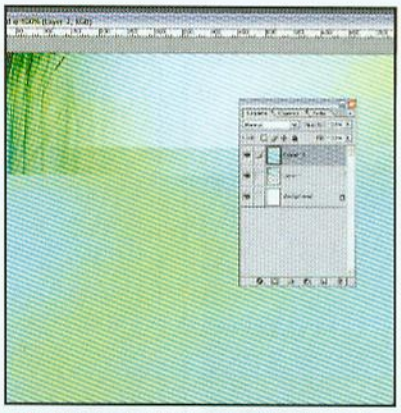
1 Mocking up a generic contents page uses similar principles to those covered in the homepage design. Create a new file in *Photoshop* using the same width dimensions of 750 pixels but leaving room for more vertical space – 1000 pixels high should be fine.



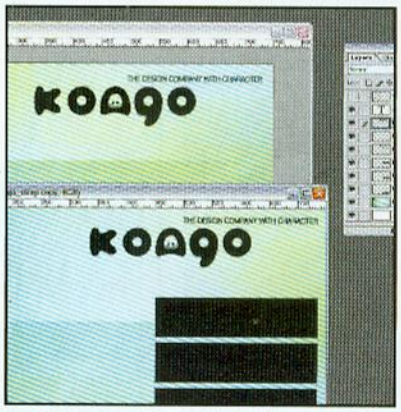
2 Keep your original homepage layout open so that you can reuse elements within your page layout. Drag a copy of the background from there onto your new canvas holding the Shift key to centre the new layer. Still holding the Shift key to retain the vertical line, drag it to the top of your canvas.



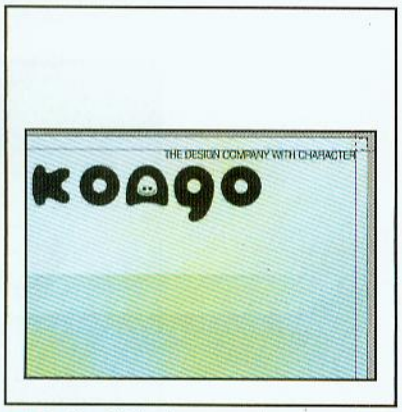
3 It would be inefficient to replicate the original complex background in an HTML page. However, you can use the design as part of the page header with a solid fill for the dominant background. Create a new layer and take a colour sample of the bottom-right corner, making sure the Only Web Colours option is *not* selected.



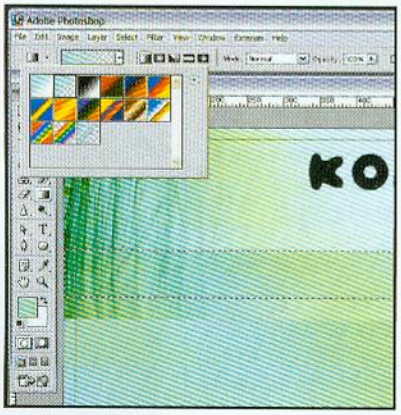
4 Fill the layer with this colour (#BEE7E1) and drag down, again holding the Shift key to retain the vertical axis, to just underneath the background's embedded gradient bar. This might work as a possible header area, although at around 210 pixels in height it might be a little too generous.



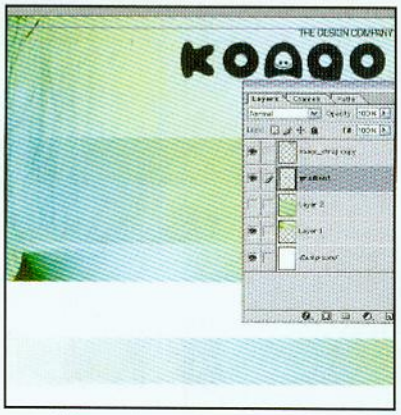
5 To test this out, return to our homepage design and duplicate the logo and strap layers. Link these together and choose to Merge Linked Layers through the layers options. You can now drag this onto the new page layout knowing that their positioning is perfectly retained.



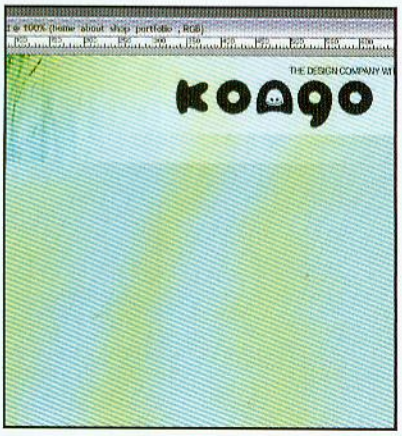
6 Position this by dragging guides 15 pixels from the page edge (as you did with the homepage) using the fixed size Marquee tool. This demonstrates that our header area is indeed too large, but we can't move the nice gradient feature because it's embedded into the background.



7 To emulate this feature on a new layer, define a selection of similar dimensions, sample the colour to the right of the gradient and choose the Gradient Fill tool. In the options bar select the linear gradient and use the drop-down to select the fill option that leads into transparency.



8 Fill the new layer's selection by clicking and holding to the right of the selection. Hold down the Shift key for a perfect horizontal movement and drag to the left of the selection. On release, you'll have a new layer with the gradient effect that can be moved into position.



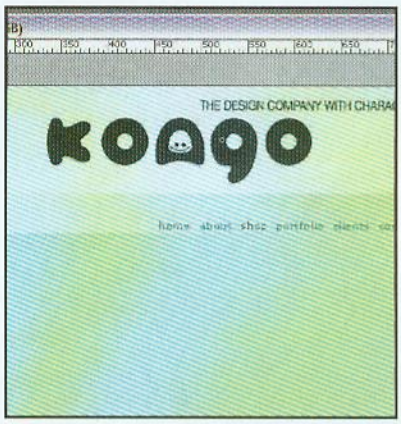
9 With your gradient layer on top, position this underneath the logo in the layout. Then position the flat-colour dominant background layer below this for the page header to start taking shape. Save the file.

Planning layouts

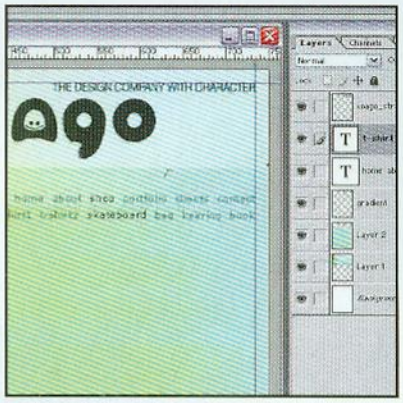
Feel free to develop design elements as you plan your layouts. However, keep within the guides you've created as these will be used to determine the HTML page layout when it comes to creating the actual pages in the next section. With a little understanding of HTML and tables, you'll be able to create flexible yet generic containers that will speed up your workflow considerably.

Early days

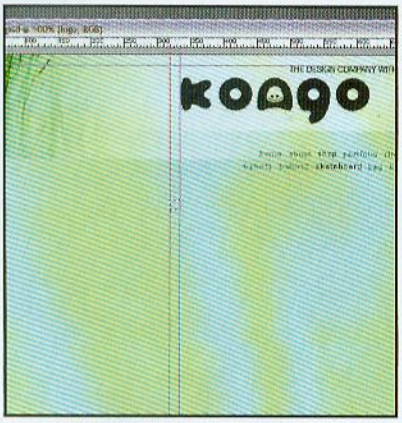
Don't forget that we're still early into the development of the site and what we're doing here in *Photoshop* is simply developing ideas from our original sketches. If you decide a layout needs drastic changes then that's fine as long as you consider how your amendments might affect other potentially reliant pages.



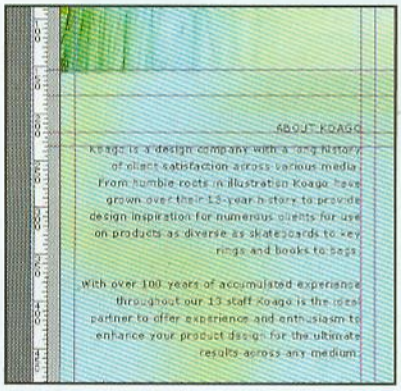
10 Drag the navigation layer from the homepage design and remove the line-breaks so that all the text is on one line. Add a 'home' link at the beginning, add a couple of spaces between each of the words and position this in the gradient bar area.



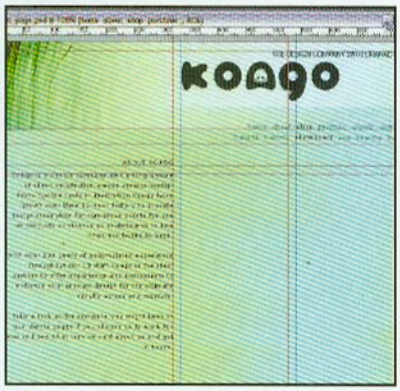
11 Duplicate this layer and drag below into the main page area. Replace the text with the options for the store product items to give our design a secondary navigation area. When it comes round to building the pages, this will be dependent on the parent page options.



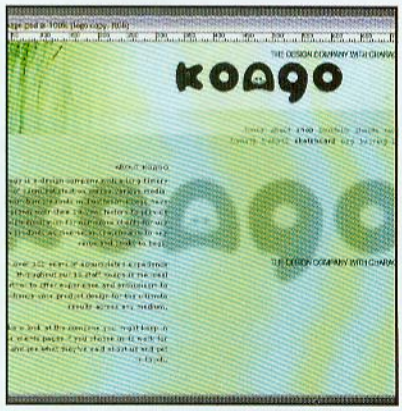
12 The next stage is to determine the best generic layout for our page content. A simple two-column approach will suffice, so drag a guide up to the logo, with a second positioned 15 pixels away to the left of this to create a central gutter.


ABOUT KOAGO
Koago is a design company with a long history of client satisfaction across various media. From humble roots in illustration Koago have grown over their 13-year history to provide design inspiration for numerous clients for use on products as diverse as skateboards to key rings and books to bags.
With over 100 years of accumulated experience throughout our 13 staff Koago is the ideal partner to offer experience and enthusiasm to enhance your product design for the ultimate results across any medium.

13 Use copy from the client's content.doc file to provide other placeholders. Copy the About Koago text from the content.doc and paste it into a text cell that's contained neatly in the left column. Add and position a title and pull-down guides to mark the positions for these elements.

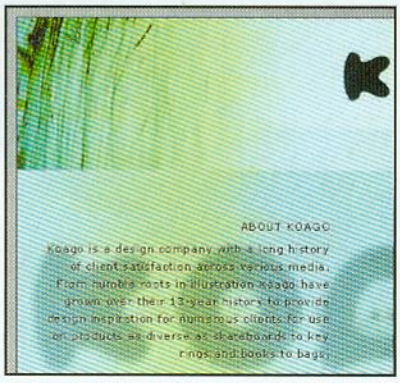


14 The right column is around 400 pixels wide and, although we may prefer to display images across this, there'll be occasions (such as shopping cart items) when we need to split this further. Using the fixed size Marquee tool, position guides to divide this area into two with a 15-pixel margin.

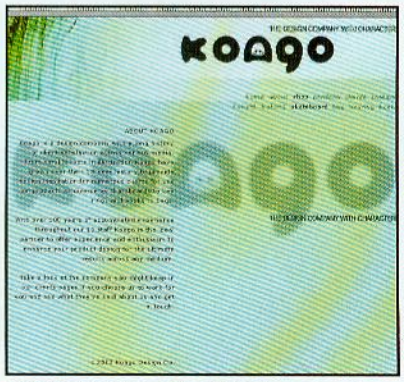


15 With our fundamental guides in place, we can continue to populate copy and images for various example pages. This is a way of making sure our basic structure considers and contains all the necessary components of our various page layouts.

16 Save the file and duplicate to try different versions to contain other areas of the site. Make sure this layout works with the template we defined in Part 2 of this section – if you need to move or add guides, try to think how they might affect other pages.


ABOUT KOAGO
Koago is a design company with a long history of client satisfaction across various media. From humble roots in illustration Koago have grown over their 13-year history to provide design inspiration for numerous clients for use on products as diverse as skateboards to key rings and books to bags.

17 Another consideration is for the placement of the search box that the client has requested. We're anticipating this will go either left in the header or to the top right of the content area, but we've left room to experiment with this when we come round to adding the feature.



18 The final element to add to our mock-up is a page footer. We're keeping things simple with a basic copyright statement that will link through to our contact page. We've right-aligned this in the left column to support further the central feeling of gravity to the page. ca-p

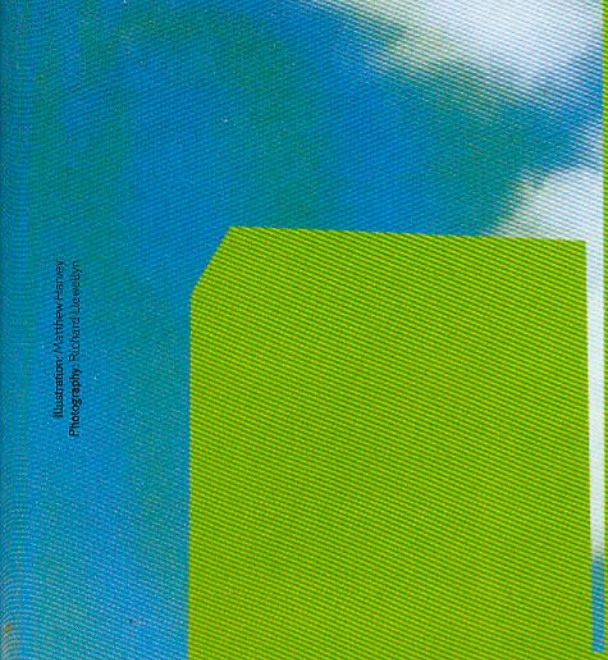


Illustration: Matthew Harvey
Photography: Richard Llewellyn

SECTION **TWO**

Creating the pages

With our designs finalised in *Photoshop*, it's time to convert them into HTML pages within *Dreamweaver*...

Converting a design from your preferred image editor into an HTML page can be done in various ways. Most applications have provisions in place for specifying slices that will constitute HTML tables. Although this process generally works fine for designers with a basic understanding of Web authoring, it often involves a lengthy and tedious configuration process. The resultant pages will do the trick but might not consider more advanced or efficient page construction.

For this reason, we're going to do things the manual way by making use of the guides we have in place to reconstruct the design as HTML tables. We'll make use of the page's background and table cell background and foreground elements in a way that even the most advanced slicing feature couldn't anticipate.

We'll be creating the pages with our ultimate *Dreamweaver* templates in mind for the most efficient layouts, allowing us to reuse elements easily and make global updates to the site through the updating of a single file. This way, our site, which needs to be completed in a matter of days, has a better chance of being completed reliably and on time.

Continuing from Section One, we'll take our *Photoshop* files and use the application's Save for Web tool. This is one of the most efficient (and easy to use) conversion tools for optimising images for use on the Web, and it provides consistently high quality graphics at incredibly small file sizes. This tool is something that every designer creating output for the Internet should be familiar with.

You'll find all the *Photoshop* files on the CD if you're jumping into our project at this point. You can still use these files if you prefer to work in *Fireworks* or an alternative image editor, but you may need to adapt the various stages.

The components needed for this tutorial can be found in the Tutorial \Section 2 folder on the CD

Part 1: Creating assets

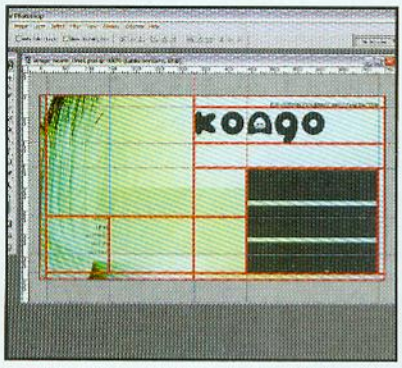
Make individual graphics from the final designs...

Image formats

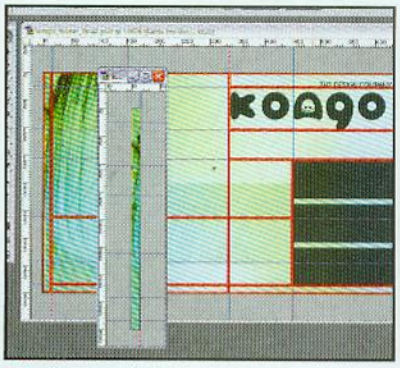
Embedded online images generally fall into either GIF or JPEG formats, and knowing what format suits which kind of image best is crucial for the most efficient creation of your site's assets. As a rule, use the GIF format for images that are composed primarily of flat imagery, such as titles and buttons, while JPEGs are better suited to more complex images such as photographs or textured backgrounds. *Photoshop* allows you to toggle between the formats in the Save for Web tool, so if in doubt, try them both to see which creates the best results at the lowest file size.

Duplication

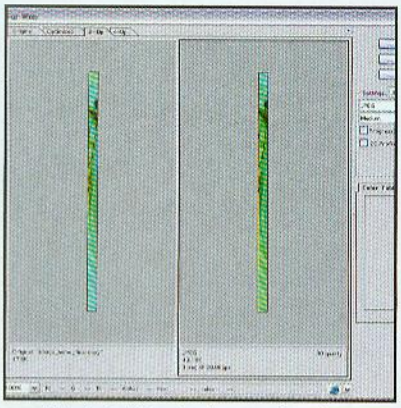
When cropping and converting original design files, it's highly recommended to work on duplicates to reduce the risk of making changes that can't be reversed. At virtually every point throughout this stage, you should use duplicate files so that if errors occur, you can simply ditch the current stage, revert to a new duplicate of the original file and start again.



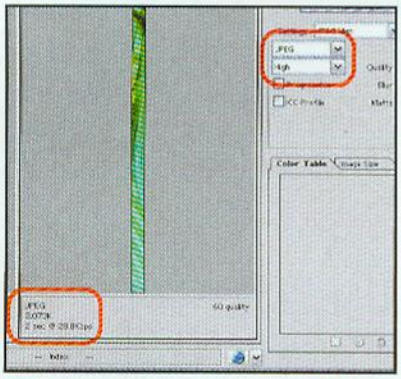
1 Open the koago_home_final.psd file from the CD and you'll notice that we've conveniently highlighted the guides showing how we anticipate the page will be recreated as an HTML table. Toggle the table borders layer on and off as required throughout these steps to remind yourself of the page structure.



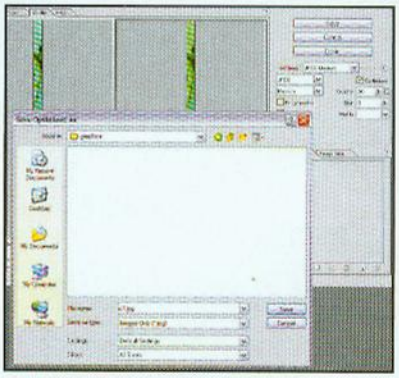
2 Duplicate the file (see the margin note) and, using the available guides, crop the left-hand vertical cell. Although this might have been included with the cells immediately to the right, it's generally good practice when layouts rely on imagery to establish the height through a single graphic in this way.



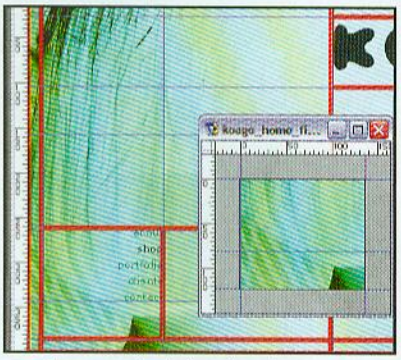
3 Hide the table borders layer and launch *Photoshop's* Save for Web tool via File>Save for Web. Choose the 2-Up view tab at the top to see the original along with a preview of your optimised version.



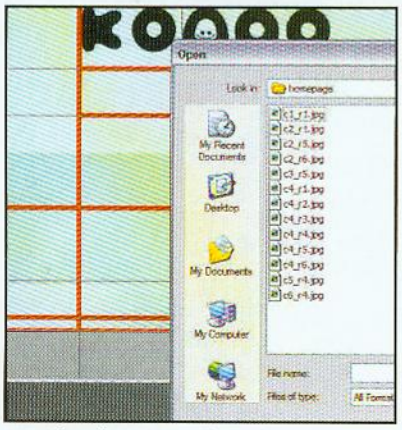
4 This variable image is best suited to the JPEG format, so choose that from the format drop-down and the preview is quickly updated. Experiment with the quality settings, paying attention to the resulting file size displayed at the bottom of the preview pane.



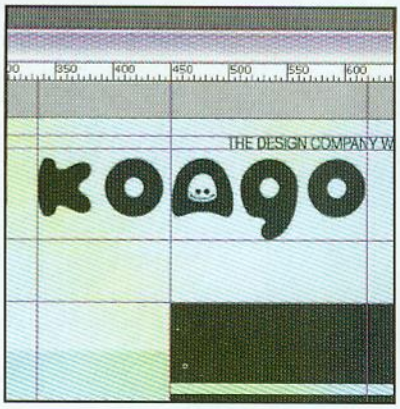
5 As the Medium setting provides a file at half the size of the High setting with no discernable difference in quality, choose Medium and leave the other settings at their default. Click the Save button to save the optimised image as c1_r1.jpg to a location on your computer.



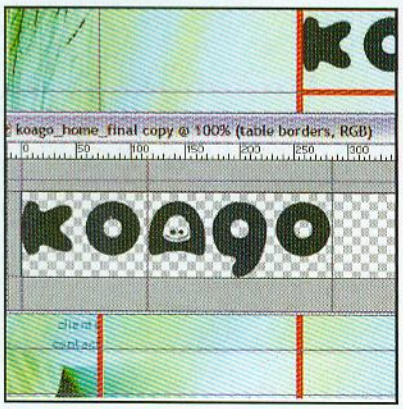
6 The name refers to the column/row coordinates – giving layout graphics such logical names provides a recognisable order when recreating the table in *Dreamweaver*. Recreate the previous steps on the other table cells. Take care to hide the text, logo, strap and image placeholders in their respective cells.



7 After a few minutes of going through the process for all the cells, you should have 13 images that constitute your background with names that identify their positions within the table. These will recreate the image as an HTML page.



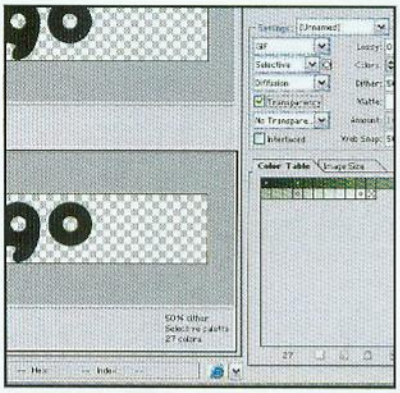
8 We didn't include the text and image placeholders because these will be separate objects within the actual page build. However, the logo and strap can be converted, but these are best saved as GIF images that will be laid over the background image separately.



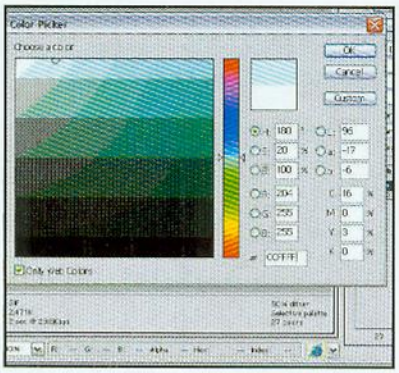
9 The GIF format supports transparency allowing this effect. As before, duplicate the file and crop the area containing the logo. Take off, and make a note of, the dominant colour background (#CCFFFF) before hiding any background layers to leave only the logo visible.

Saving images

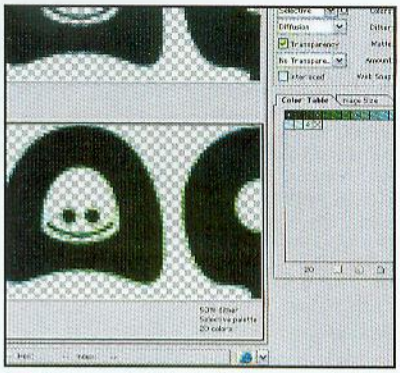
When saving your prepared images before building your pages, keep them together. Create a graphics or images sub-folder within the folder that will contain your HTML files. Images for a unique page, such as the homepage, can be further organised into a subsequent folder for easy access and to avoid getting them confused with other pages. You can have as many folders as you want, and organising your assets in this way will help you to manage your site efficiently in the longer term.



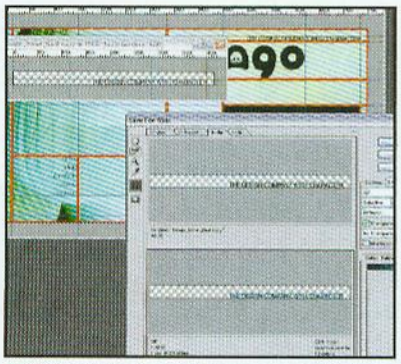
10 Launch the Save for Web tool and change the format to GIF making sure the transparency option is checked. As our logo and dominant background colour are Web-safe, there's no need to alter the Dither or Web Snap options.



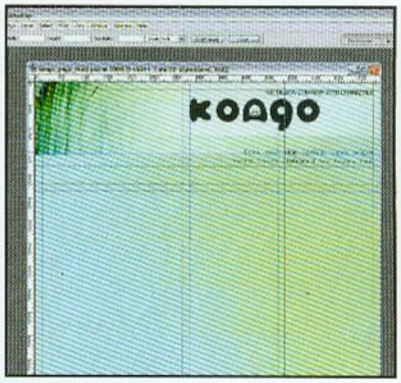
11 Use the arrow by the Matte field to choose the Other option and enter the colour code for our sampled dominant background (#CCFFFF). Click OK and zoom into the preview to see the anti-aliasing around the image, which will smooth the blend into the background.



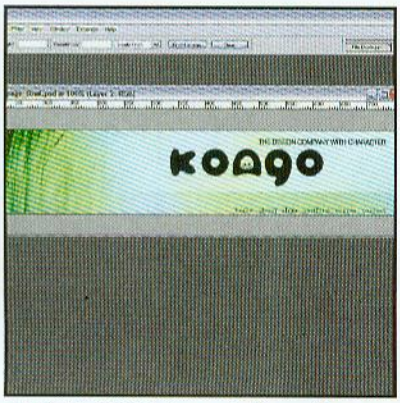
12 Paying careful attention to how your selection affects the preview image, knock down the amount of colours used to the least possible. We've settled at 16, which may seem extravagant but our logo file size is now just over 2KB and the excess colours will provide smoother anti-aliasing.



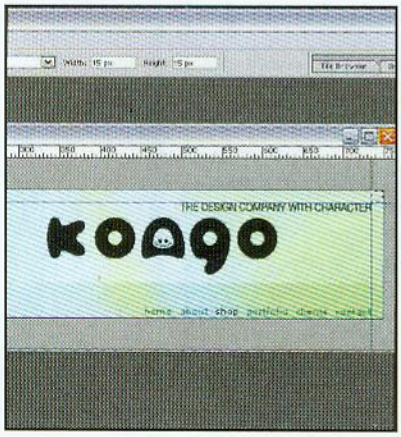
13 Click OK and repeat the process with the logo strapline, saving that too as a GIF with transparency using anti-aliasing to blend into the background. You could crop closer around the image, but the reduction in file size is negligible and you may run into alignment problems when building the page.



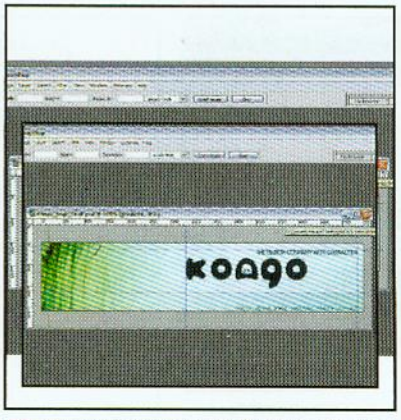
14 Similar processes can be used against the generic page design. Open the koago_page_final.psd file that you created in the previous section – there's a copy on the CD if you need one. This provides the opportunity for you to determine how the imagery will need to be sliced.



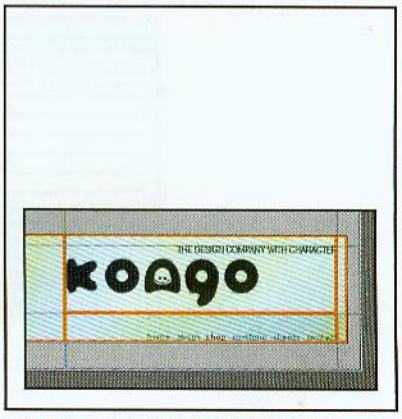
15 As the HTML background will cover the page colour, we only need to concern ourselves with the page header. Duplicate the file and remove the current guides via View>Clear Guides and crop the header neatly under its image to isolate it.



16 Start placing guides by introducing our 15-pixel border to the top and sides of the header. As the HTML page will include this border, the inclusion of it will allow us to rebuild the page into a single, more efficient table.



17 The next logical divides to impose on the page are to the left of the logo. With the layer selected, drag a guide until it snaps into place. The gradient layer also provides a convenient point for another horizontal guide, leaving your header looking something like the above.



18 Try to figure out for yourself how this might best be sliced and check out the koago_header_final.psd file to see how it compares with ours. All that's left now is to crop each cell and run through the Save for Web tool as we did with the homepage graphics.

Part 2: Building the page

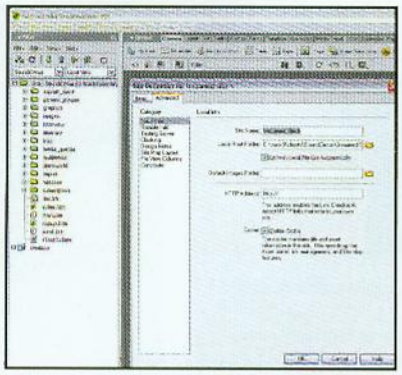
Put the assets back together into an HTML page to recreate the layout...

Are you local?

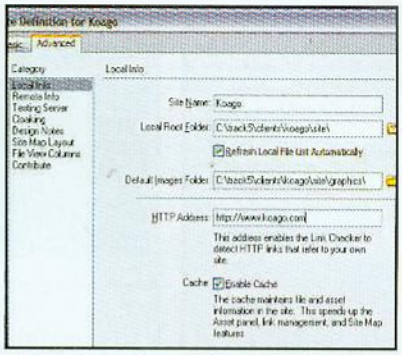
You will hear frequent mention of the words 'local' and 'remote' when working on online projects. This refers to the relative location of any files that you may be working on. If something is 'local', then it's stored in a folder or directory on your own computer, while 'remote' refers to the server that ultimately hosts your site.

Using your workspace

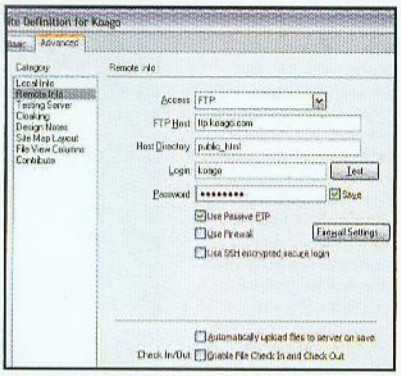
If your screen resolution allows, you might choose to move your Site and Assets panels over to the left of *Dreamweaver's* interface. This will free up space on the right, making more panels available while giving ample space on the left for you to have constant access to your site pages and their graphics.



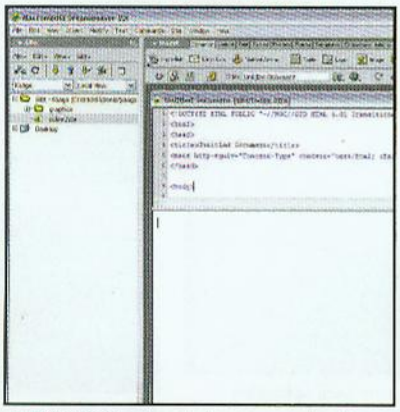
1 You should now have the images required to recreate the page layout. Take these from the CD if necessary and save them to corresponding folders within a parent graphics folder in which your site pages will be held. Launch *Dreamweaver* and choose Site>New Site.



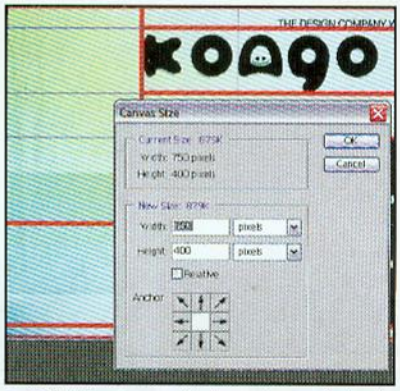
2 Use the tabs to move into Advanced mode, select the Local Info category and enter the name Koago. For the Local Root Folder and Default Images Folder locate the folders in which your site and its graphics are contained. Enter the site address as http://www.koago.com and leave the cache enabled.



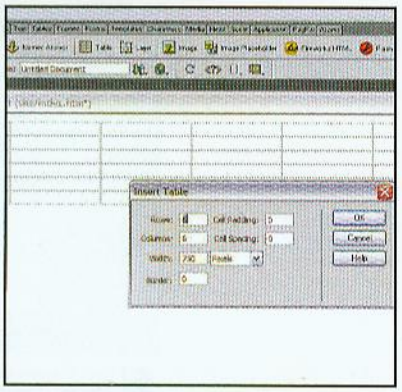
3 The Remote Info category will eventually need completing to upload the site with FTP address, username and password details. For the purposes of this tutorial, though, you can leave the Access option set at None to continue working on the site locally.



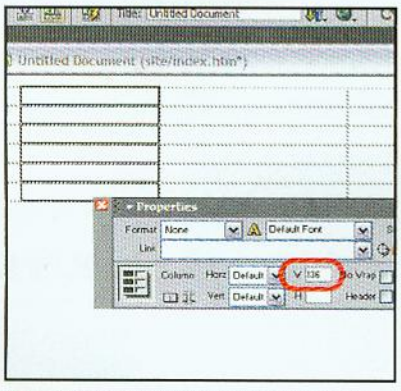
4 Click OK to return to *Dreamweaver's* main interface and you'll be presented with a rather sparse range of options. Create a new file in the Site panel and name this: index.htm. Open this and we're ready to start creating our homepage.



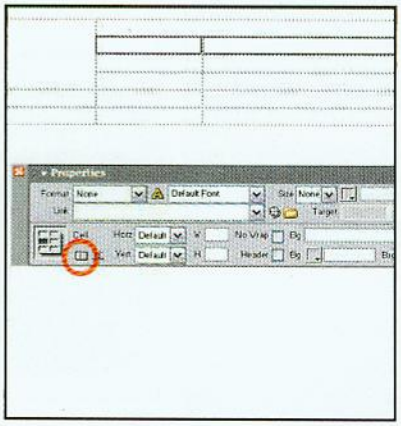
5 First though, open the koago_home_final.psd file in *Photoshop* to refer to our planned table layout. Make sure the table borders layer is visible and make a note of the number of rows and columns required (6x6) and the width of the canvas (750 pixels).



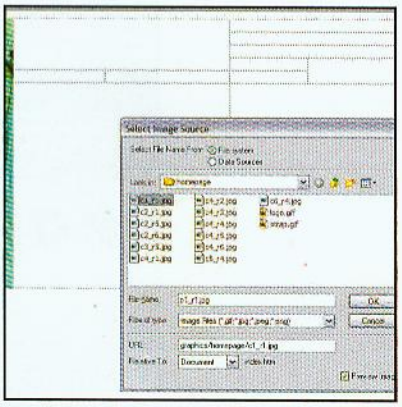
6 Returning to *Dreamweaver*, create a new table via Insert>Table and use these attributes to determine the table properties with 6 rows, 6 columns and 750 pixels in the Width field. Leave the Cell Padding, Spacing and Border options at 0 and click OK to generate the table.



7 Use the Marquee tool in *Photoshop* to have the width of the columns displayed in the Info panel. We calculated 15, 136, 184, 115, 285 and 15 pixels respectively, which can be applied to the HTML table's columns by dragging to select an entire column and entering the dimensions into the width field.



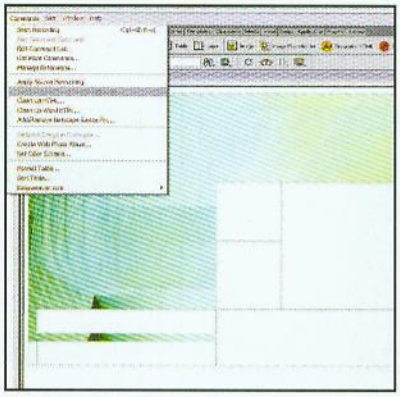
8 You can now recreate the PSD file's cell structure in *Dreamweaver*. To do this, click and drag the corresponding cells until they're highlighted and use the Merge Cells option found on the Properties panel to group them together.



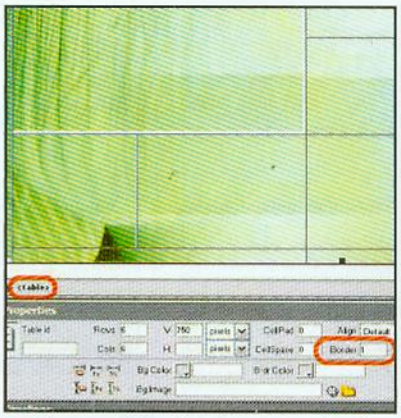
9 The height will be determined by the contained images, so don't worry if things are looking a little squashed at the moment. To insert an image, simply place the cursor in the relevant cell and select Insert>Image and browse to the required file within the graphics/homepage folder.

Source formatting

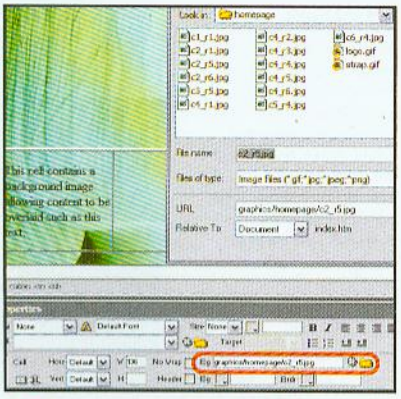
There will be occasions when things don't quite appear as anticipated. Often, this is nothing to do with errors on your part but rather to do with *Dreamweaver* trying to anticipate what your actions are doing to the HTML code under the bonnet. If your layouts look disjointed, use the Apply Source Formatting option under the Commands menu and things should jump into place.



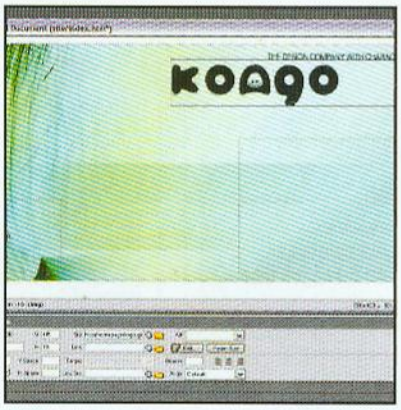
10 Do the same for the other cells to recreate the original design within the HTML table. Make sure you use only the background elements as opposed to the logo and strap gifs. Apply Source Formatting (see margin note) if things look jumbled as you build the page.



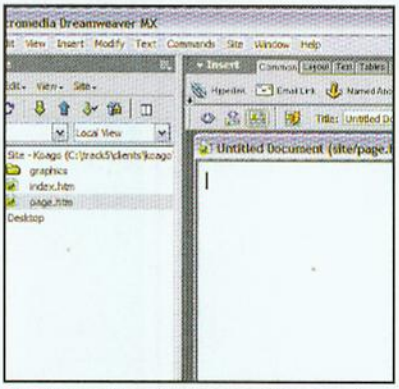
11 As some of the cells need content such as the navigation, logo and images overlaid, we'll need to move areas of the design into the background. To make things easier, select the <table> tag and make the table borders visible.



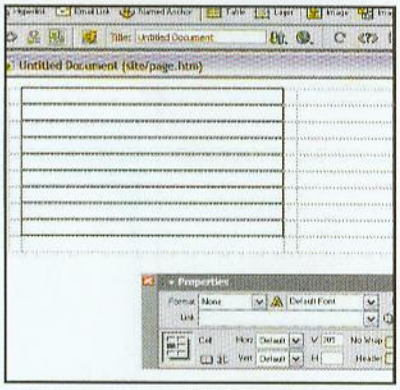
12 In the relevant cells, delete the image currently in place and instead put them into place using the background field, which becomes available when the cell is active. You'll need to do this to the cells that we've planned to contain the navigation, logo, strap and image placeholders.



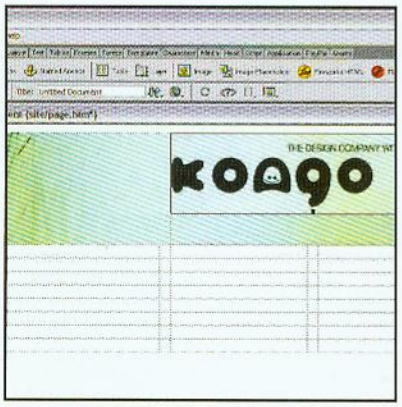
13 With the background now configured, you can insert the images into the appropriate cells for the logo and strapline. Enter the text for the navigation too using Shift-Enter for regular line breaks. Once complete, turn off the table borders.



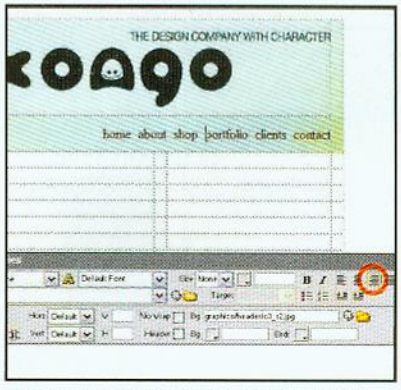
14 Save and close the homepage file and create a new page called: page.htm. Open this to start working on the generic page layout. At the same time, jump over to *Photoshop* to remind yourself of the layout using the appropriate page and header files.



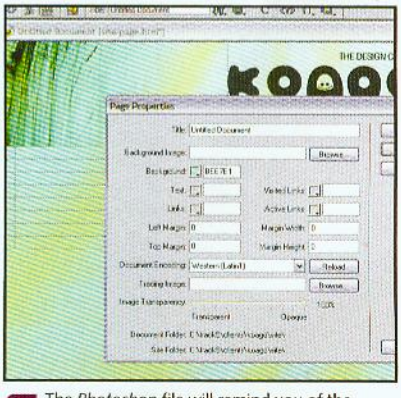
15 Recreate the page table by inserting a table of 750 pixels using 7 columns and 10 rows. We're deliberately using more rows than necessary as any excess can be removed once the page has been completed. Specify the column widths to 15, 305, 15, 192, 15, 193 and 15.



16 Use the top two rows to merge cells to recreate the koago_header_final.psd table-border-layer guidelines. Then insert the images as we did with the homepage, including making use of the background level with the logo on top, for example.



17 Ctrl-click (Win) or Command-click (Mac) the table cell where the primary navigation is to appear. Choose to right-align the content through the Property panel before typing in the necessary text. Use Ctrl-Shift-Space (Win) or Alt-Space (Mac) to enter extra spaces between words.



18 The *Photoshop* file will remind you of the dominant background colour (#BEE7E1). Choose Modify>Page Properties and use BEE7E1 in the background field. Enter a value of 0 into each of the Margin fields to lose any spacing between your table and the browser window. Click OK and save.

Part 3: Formatting the text

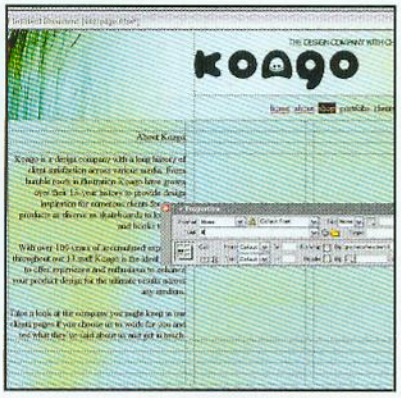
Create a linked Cascading Style Sheet (CSS) to control the appearance of the text...

Empty links

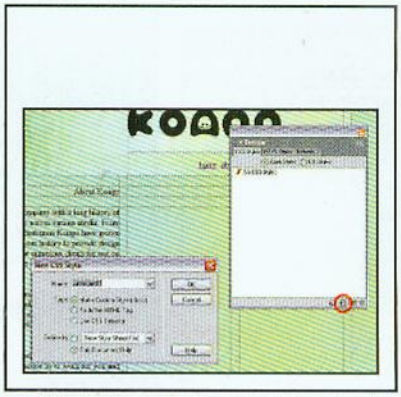
To emulate a link when there's no page to point to (as we're doing here to define the CSS link properties) you can enter a hash symbol (#) as the link location. This can also be used to generate link events that may be necessary for such behaviours as pop-up windows – in this case, you'd create an empty link and associate and configure the relevant behaviour from that.

Style editor

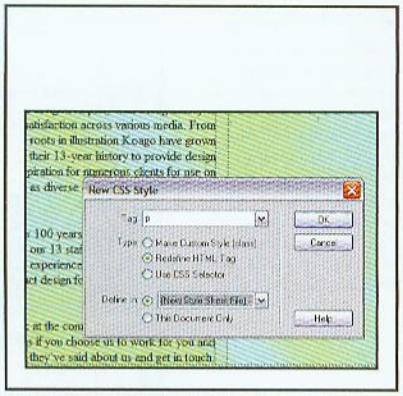
Dreamweaver provides comprehensive tools for working with CSS files, although the Wizard approach it provides can become rather limiting for more heavyweight work. Dedicated CSS editors such as *TopStyle* have intuitive interfaces that allow much more control over various aspects of the code. Plus, there's instant access over more specific features that would be impractical to integrate into less specific titles.



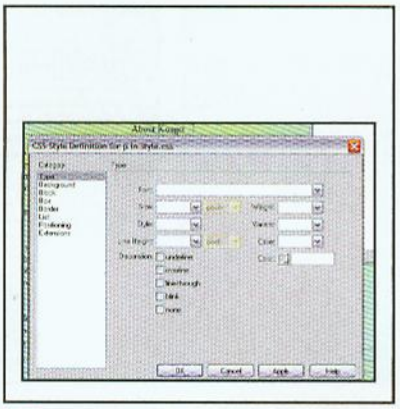
1 Open the page.htm into which we'll add some temporary text. Take the copy from the content.doc file and enter into the appropriate cells, which should then be right-aligned. Select each of the navigation words and create an empty link by associating a hash (#) symbol as the link location.



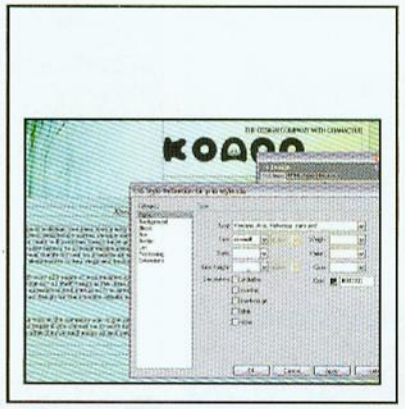
2 We now have a number of basic elements that can be formatted through CSS. Open the Design panel and access the CSS Styles tab. Along the bottom of the pane are four icons for controlling your CSS files; click the New CSS Style option to start the process.



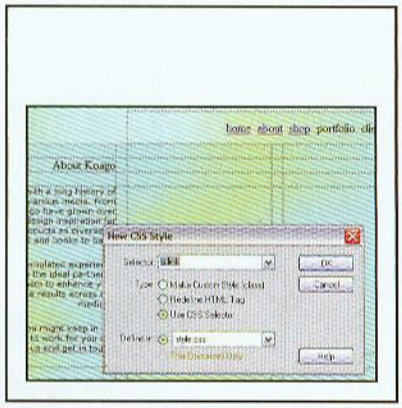
3 In the New CSS Style dialog box, select the Redefine HTML tag option and choose the paragraph tag (p). Before clicking OK, choose to define the CSS file as a New Style Sheet File; this creates an external .CSS file that can be linked from other documents in our site.



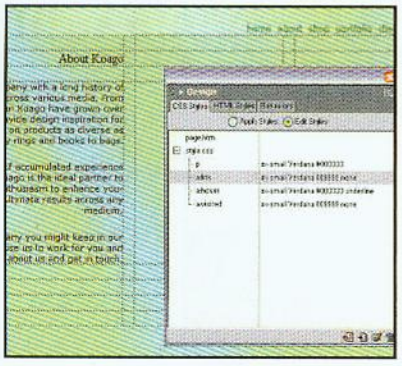
4 Click OK to specify the file name and location of the created file. Call it style.css and save to the root of your site. When this has been done, you're presented with *Dreamweaver's* CSS Style Definition interface from where you can configure the appearance of your site's text.



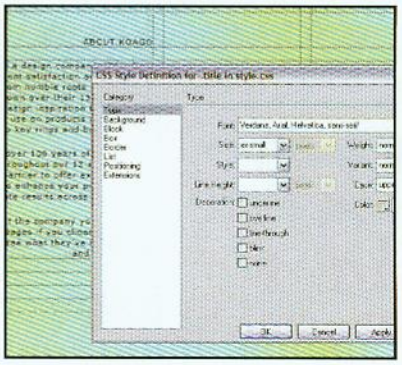
5 Choose the Verdana font, size xx-small and use the colour sampler to take the font colour from the page logo (#003333). Click Apply and you'll see how your changes affect things before you commit. Click OK when you're happy with the results.



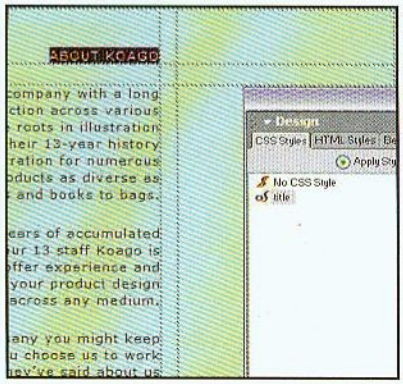
6 To amend the link behaviours, return to the CSS Styles within the Design panel and click the New CSS Style button again. This time choose the Use CSS Selector option to make the various link states available under the Selector options. Find the a:link and click OK.



7 You can now determine the properties of your up-state links as before, except you'll have options for various decorations – which we've set to none – associated with the links status. Repeat these steps for the a:hover and a:visited states using the *Photoshop* design for their values.



8 For the page title, we'll use a CSS class so we can apply styles to selected text. Create a new style, this time opting for the Make Custom Style (class) option and entering the name: .title. Click OK to once again format the font details before clicking OK to return.



9 You can now select text directly on the page and click your newly created CSS class to determine its appearance. Experiment with other CSS styles and take a look at the final style.css file that we developed for the site. **ca>p**

SECTION **THREE**

Working with templates

With *Dreamweaver* came templates, and with templates came a revolution in the way Websites were designed, created and managed – suddenly, the creation of online content wasn't that tricky after all...

When *Dreamweaver* first emerged, there were a number of key features that made the application stand out from the new movement of WYSIWYG Web-authoring platforms. *FrontPage* had taken the lead at the time and, having close integration with the already well-established Office tools, the application certainly had a head start when compared with the competition. Macromedia and Adobe soon realised the potential of this new method of creating online content with their equivalent titles, but it was *Dreamweaver* that met the challenge and took on the might of the real contender: the HTML hand coder.

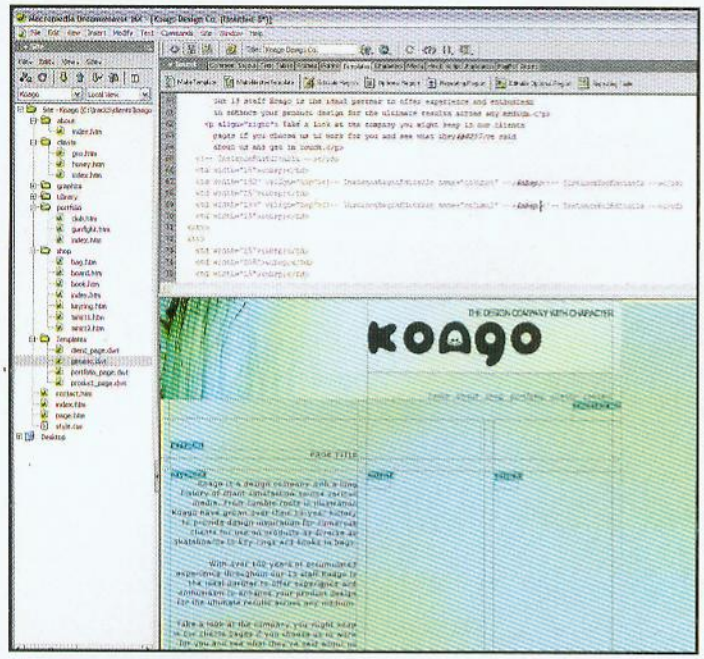
A rare beast that rarely spoke except to snort coffee at the sight of *FrontPage* source code, the HTML coder had made the creation of online content an impossible task for a regular designer. And when *Dreamweaver* emerged there was outrage as the application actually claimed to create clean code and managed entire sites intelligently. Worst of all, *Dreamweaver* introduced the concept of templates, making the creation and management of large scale sites a much more capable task for even the humble pixel-pushing designer.

Nested templates are a new feature to *Dreamweaver MX*. They provide a method of creating unique templates based on an existing parent template file.

A new age dawns

Templates allow the creation of a single page that contains and controls key components of generic Web pages. The concept dictates that once a template is created, any number of child copies of the page can be generated to contain their own unique content dependent on >

Illustration: Matthew Harvey
Photography: Richard Llewellyn



With templates in place, only regions predefined as editable can be amended by the author. Even code outside these areas is greyed out and disabled from hand coding.

their purpose within the site. Should the parent template be changed in any way, such changes will be automatically inherited by all reliant pages.

The idea was to cause a storm in the development and production of online content as it no longer meant hours or even days of hand-coding tedium. To change a corporate logo, for example, meant a simple single amendment to the global template, which would be instantly applied to subsequent pages. The revised site could then be uploaded with its new appearance in a fraction of the time.

Dreamweaver's approach had additional benefits for site managers: templates could be created for use on large corporate sites where multiple authors could now update content.

Macromedia contribution

Web authoring for the masses...

Creating a client-managed site through *Dreamweaver* might be a reasonable option, but this still requires the client to buy and figure out what is essentially an expensive pro-level development application. In a real world setting, this often leads to fear and confusion about a fundamental task.

A more realistic option might be to suggest the use of Macromedia *Contribute* software. The software allows any regular HTML site to be updated easily by anyone with basic word processing knowledge and can be learnt in a matter of minutes. A crucial advantage when being used with *Dreamweaver* sites is that original templates are recognised, but their restrictions are still in place. This means that regions defined as locked within a template are locked from editing within *Contribute*.

With restricted permissions, only the editable regions of a template would remain editable, further protecting the template areas such as page headers or navigation. Initially, the idea of multiple authors being left free to manage their own content would have filled most site managers with dread at the thought of mutilated code being returned or house style being disregarded, but protected templates made the whole process possible.

Ultra-templates

As templates and *Dreamweaver* became more established, Macromedia was keen to move into new territories as it continued to challenge the might of the coders. *Ultradev* was introduced and developed as a hardcore *Dreamweaver* variant providing additional support for database-driven sites through *Dreamweaver's* familiar and easy-to-use interface.

Dynamically driven sites tend to use a template principle inherently. A shopping cart may have a generic product page populated on the fly by content inherited through a command passed through the URL. For example, a shopping site would provide all its products through a shell page while content would be referenced and drawn from a database field that's prompted by the address that's passed on. This approach means that some of the biggest sites in terms of content can be among the smallest in terms of unique pages.

The first versions of *Ultradev* evolved from Macromedia's acquisition of a software title under the name of *Drumbeat*, and although the revised title had obvious parallels with *Dreamweaver*, there were still essential differences as to how the software functioned. However, as the application further evolved, the boundaries were becoming increasingly blurred.

MaXimum templates

By the time *Dreamweaver MX* was launched, *Ultradev* had become fully integrated into its stable mate. The difference between the titles was essentially between the dynamic capabilities and, as sites were becoming increasingly reliant on such content, it was decided that development should continue within a single application.

The benefits this had on templates were huge as *Dreamweaver* could now support templates with repeated regions that could be controlled by database content. This meant that dynamic content could intelligently control the amount of information available within a page regardless of whether there were one or one hundred items to display.

Library items also emerged to support templates, providing individual components such as navigation that could be inserted as required and updated, as with templates, via a central entry.

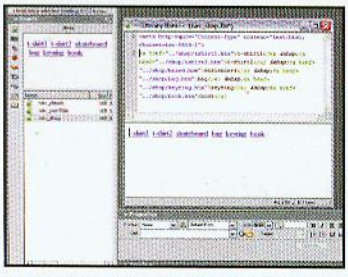
Nesting templates were another key improvement that further support the true dynamic model. A generic page can be created with nested elements that determine, using our Koago model as an example, whether the page is to function as a product or as client information.

Editable tag attributes allow more flexibility for the site manager in terms of what can be edited within a page that's controlled through a template. Previously, an object was either available for editing or it was locked, but through this new feature, specific attributes of a locked object can be amended. This is useful if, as a site manager, you don't want to allow authors to amend the dimensions of a specific contained table but you do, for example, want to allow them control over the background colour.

Like repeated regions, optional regions within a template allow you to show or hide an element based on a condition based on the template parameters. These are effectively variables set to a particular value that determines whether the optional region is made available. This is particularly effective for use within nested templates to control the way a page looks without having to create separate nested templates for each different design possibility.

Our Koago project doesn't have the scope for working with dynamic database-driven content at this point, but through good use of templates we can streamline the production of our site effectively in anticipation of future dynamic developments.

Files on disc
The files needed for this section are in the Section 3 folder on the CD.



Library items are an incredibly useful partner to templates, allowing for individual components such as navigation to be controlled and updated from a central point.

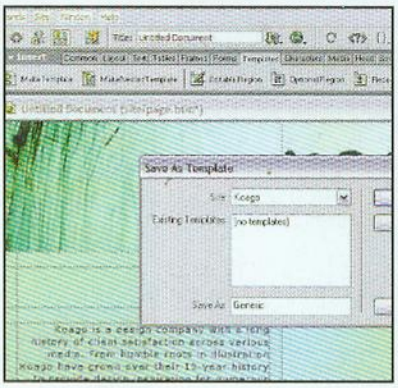
Part 1: Integrating templates

Convert your page layouts into reusable *Dreamweaver* templates...

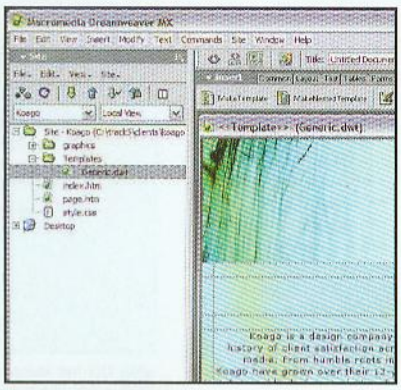
Think ahead

Careful planning is needed for template creation because once a template has been configured and used to create pages, it can be complicated to adjust later on. You can make relatively easy changes to the original file by adding new regions, for example, but taking regions away can lead to problems. Existing content will need to be relegated to another existing area if you want to keep any copy contained, but this will undoubtedly mean you having to go through each of the pages to check any formatting inconsistencies.

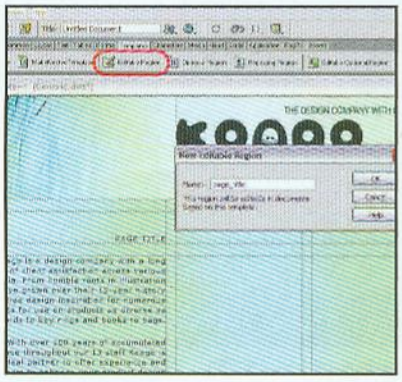
1 Open the page.htm file you created through the previous section or use the files provided on the cover CD. The layout is pretty sparse and there are a few tweaks you'll need to do before this can be used as a generic page template.



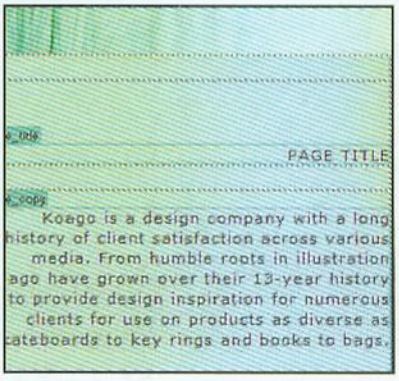
2 First, you need to specify your file to be a template, which can be done by selecting the Template tab found on the Insert bar and choosing the Make A Template option. In the dialog box that appears, leave the site as Koago and type Generic in the Save As field.



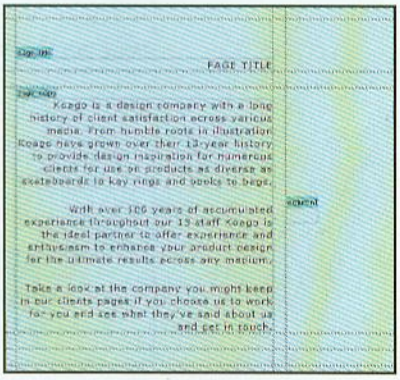
3 Click on Save and your file is converted into generic.dwt (*Dreamweaver* template format) saving itself into a newly created Templates folder within your site structure. Any subsequent templates you create are stored here; never move files out of this folder or they'll become unusable as templates.



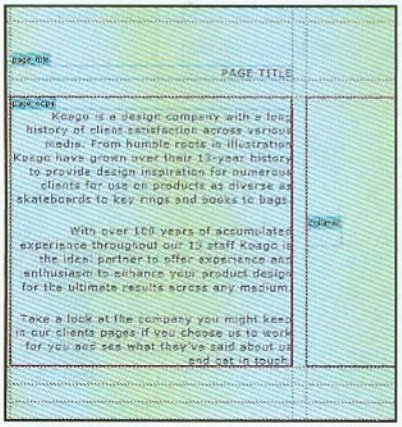
4 You can now select components of your layout that will need to be editable, such as the page title, and click the Editable Region button within the template tab of the Insert bar. This prompts you to name the region, so use something obvious like 'page_title'.


PAGE TITLE
Koago is a design company with a long history of client satisfaction across various media. From humble roots in illustration ago have grown over their 13-year history to provide design inspiration for numerous clients for use on products as diverse as kateboards to key rings and books to bags.

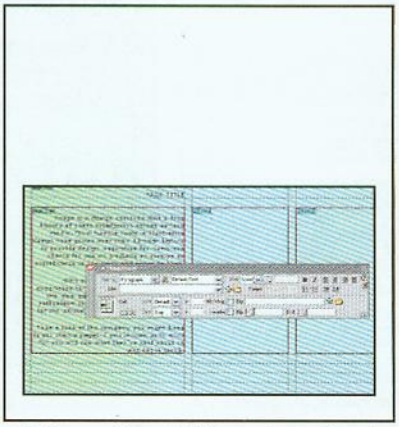
5 Click OK and repeat for the page copy and you'll notice these areas are boxed and titled for easy identification. These become the page areas that can be further edited, while other aspects such as the primary navigation and page layout are locked from access.



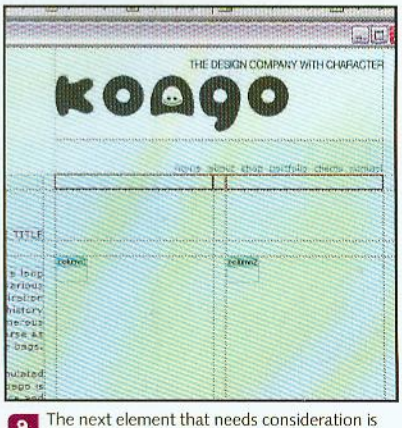
6 The two right-hand columns are available for content depending on the page function, so leave these empty. To make the cell editable Ctrl-click (Windows) or Option-click (Mac) in the cell to select it and then run through the same process to create the editable region.



7 With both columns converted, you'll notice the contents are centrally aligned vertically by default. The same goes with the main page_copy cell, but the content hides this at present. Use the same trick to select all three cells simultaneously.



8 Now use the vertical alignment option found on the Properties panel to align these to the top. This ensures that any content placed within these cells is placed neatly at the head of these invisible cells rather than in its default position.

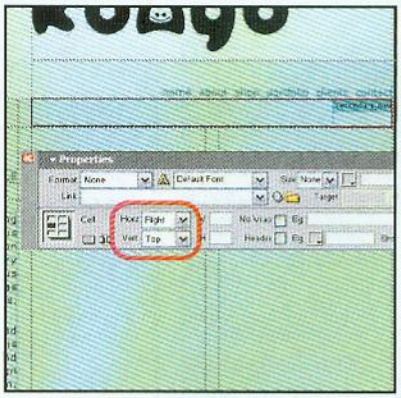


9 The next element that needs consideration is the secondary navigation. This too varies depending on the page content, but you need to provide a consistent area within your template. Click and drag through the three cells below the primary navigation cell to select them. >

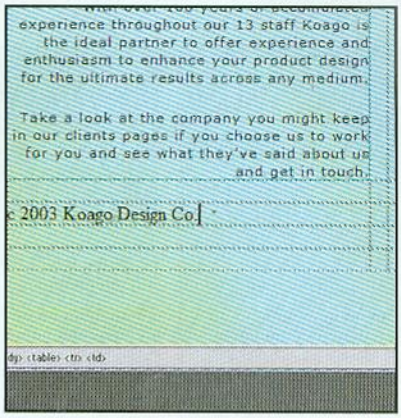
Integrating templates continued...

Multiple templates

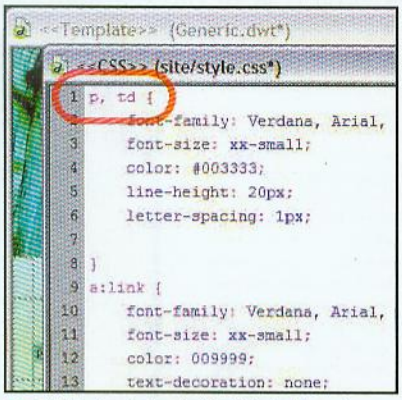
Don't feel confined to a single template. We only have time to work on one in this project, but many sites will use multiple templates for different areas. We could have used other templates for areas such as the main product page to present information in a different layout. However, this means that any site-wide changes you want to make, such as updating your copyright information, will need to be applied to all your templates before they can be made effectively.



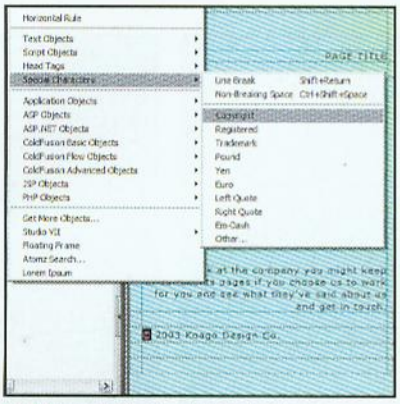
10 Use the Merge Cells button on the Properties panel and create a new Editable Region as you did before by selecting the table cell before running through the process. Leave the cell's content empty and name this secondary_nav. Right- and top-align the cell to match our original designs.


experience throughout our 13 staff Koago is
the ideal partner to offer experience and
enthusiasm to enhance your product design
for the ultimate results across any medium.
Take a look at the company you might keep
in our clients pages if you choose us to work
for you and see what they've said about us
and get in touch.
© 2003 Koago Design Co.

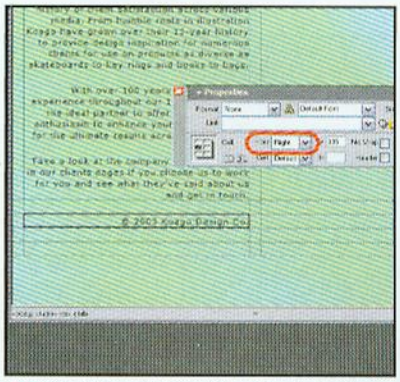
11 We can now add our page's copyright statement. Position the cursor in the relevant cell below the page copy and type in the required text. Notice how the text is rendered with its default values – this is a problem within our style sheet.



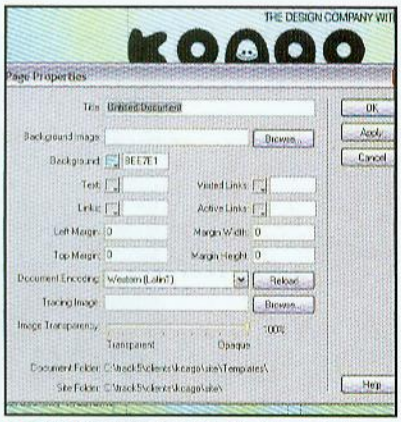
12 This is due to the fact that the text is *not* contained with paragraph tags but directly within the table cell tag <td>. Open the style.css file directly from the Site manager panel and adapt the code as above adding the td separated by a comma from the existing paragraph tag <p>.



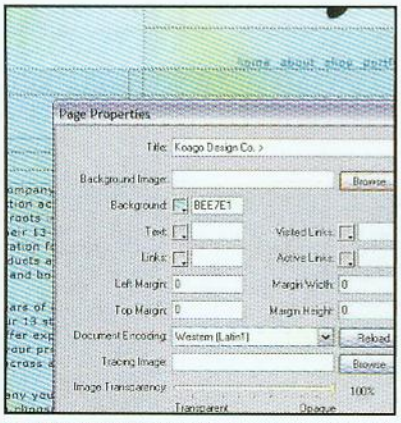
13 Save and close the file to return to our template and you'll notice the text has now reformatted itself to become consistent with our design. Enter the actual copyright symbol by deleting the temporary 'c' character and choosing Insert>Special Characters>Copyright.



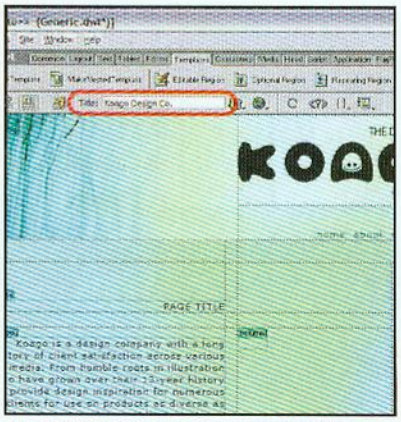
14 Select the containing table cell and align the contents to the right to further match the original design. There's no need to convert this into an editable region as this will remain unaffected by page content, but should revision be necessary in future we can change the template to update reliant pages automatically.



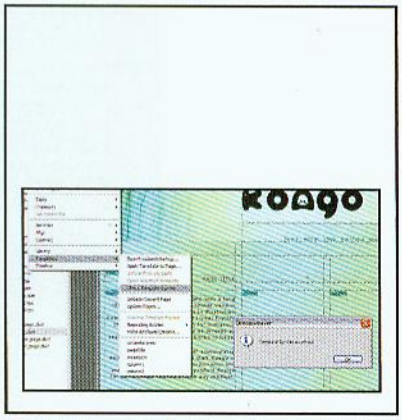
15 With the basic template virtually complete, we only need to change the page title to contain the same initial text when new pages are created. Choose Modify>Page Properties to launch the dialog box for options related to the page in general.



16 The Title field determines the content visible in the top bar of the actual browser window and provides various functions to both search engines and the local browser. Ultimately, these need to be unique to each page but, for the time being, simply enter: Koago Design Co.



17 Click OK and you'll notice the same Title content is also available through the document information accessible via the top of the page. Once you get round to creating specific pages, you'll be able to make Title changes directly from here without needing to go into the Page Properties.



18 Finally, with the basic template complete it's good practice to check the template syntax before moving on to discover any potential problems. Choose Modify>Templates>Check Template Syntax to run the test and save the file when any problems have been sorted.

Part 2: Advanced template features

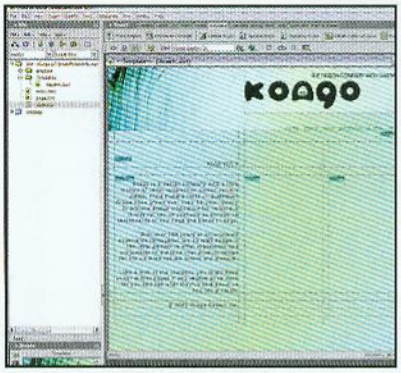
Add features to create more efficient templates...

Use the library

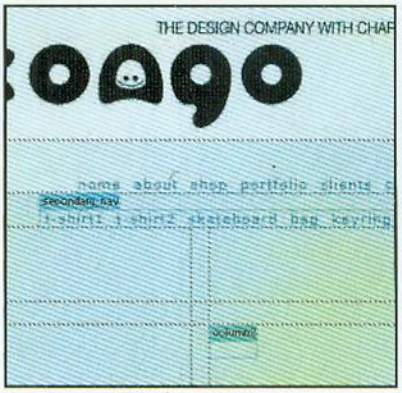
Library items provide the option to save components of a Web page into a centralised location that can be updated (not unlike complete template files) to affect multiple pages that rely on that particular object. Library objects are saved with the .LBI extension within a created Library folder – these can then be opened individually, modified and saved to instantly update any reliant pages.

Asset management

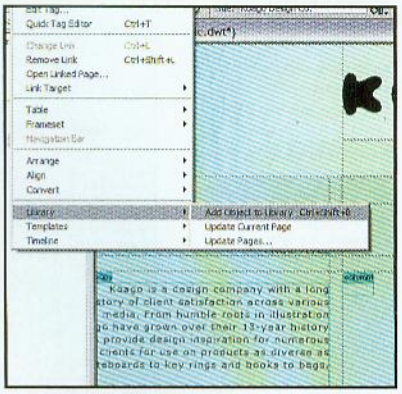
Templates and Library items are all available centrally through the Assets panel, which also provides access to your site's images and *Flash* files along with colours and links your site may use. Keep this open and get into the habit of using this to improve your workflow.



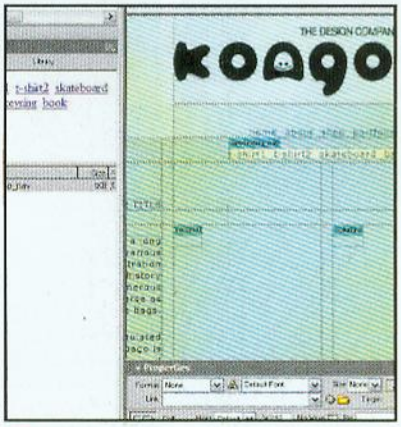
1 We could now go straight into the production of our site, but there would still be a lot of duplication involved within common areas. To get round this, we're going to further develop our templates. Open the generic.dwt file you completed in the previous part.



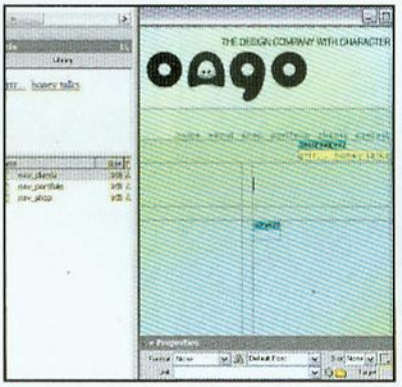
2 In the secondary_nav region create empty links using the # symbol for the shop pages of t-shirt1, t-shirt2, skateboard, bag, keyring and book. We're still using empty links as we have no pages currently created that the links can point to.



3 Select the newly created navigation and choose Modify>Library>Add Object to Library. Ignore any alert messages warning you about Style Sheet information because this will be covered by the page's main CSS link reference. Click OK.

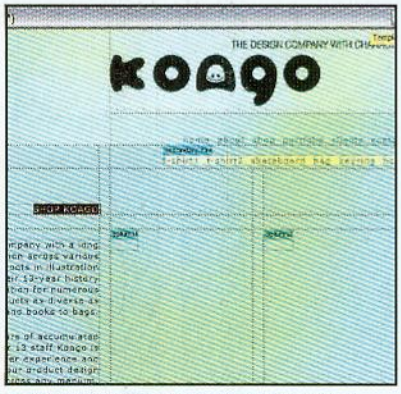


4 Open your Assets panel if it isn't available (via Window>Assets) and you'll see your newly created Library item available as an object. Rename this to nav_shop and choose *not* to update any template pages at this point.

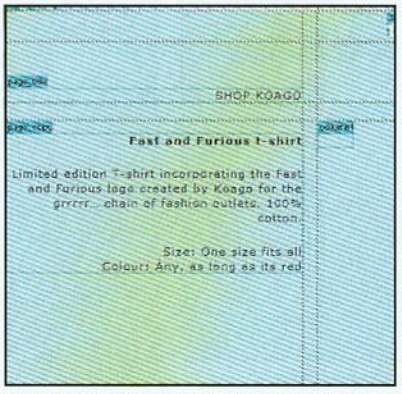


5 Your shop page's secondary navigation is now saved in *Dreamweaver's* Library ready for use when we create these pages. Select the Library item from our main template and delete. Repeat the process for the clients and portfolio sections to create secondary navigation elements for these pages.

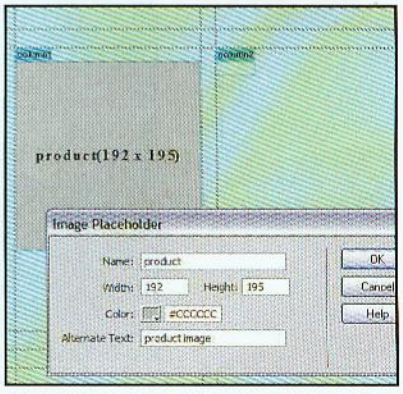
6 Creating a nested template provides a method of modifying our generic template to include specific entries to a particular area such as the shop while still retaining global access to other reliant pages. Create a new page (File>New) using our generic template.



7 Open the koago_product_final.psd file to remind yourself of the intended generic product content. Back in *Dreamweaver*, drag a copy of the nav_shop Library item into the secondary nav and change the page title to read SHOP KOAGO to match the *Photoshop* design.


SHOP KOAGO
Fast and Furious t-shirt
Limited edition T-shirt incorporating the Fast and Furious logo created by Koago for the grrrrr... chain of fashion outlets. 100% cotton.
Size: One size fits all
Colour: Any, as long as its red

8 Next, copy the text from the left-hand column of the *Photoshop* file and replace that which is currently being held as a placeholder. You'll need to reformat the text slightly to provide the correct line-breaks and embolden the product title to match our *Photoshop* design.

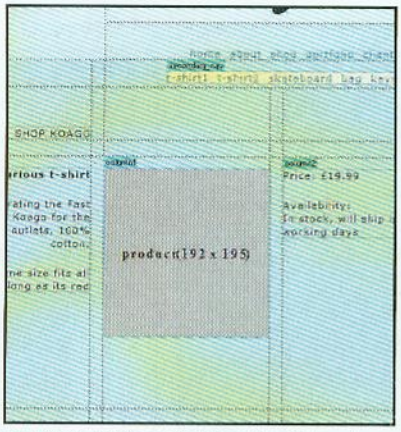


9 Place the cursor in the central column, which we've named column1 in our template, and delete the non-breaking space currently populating the cell. Then, insert an Image Placeholder via the Common tab of the Insert panel. Enter dimensions of 192x195 pixels and other info as above.

Advanced template features continued...

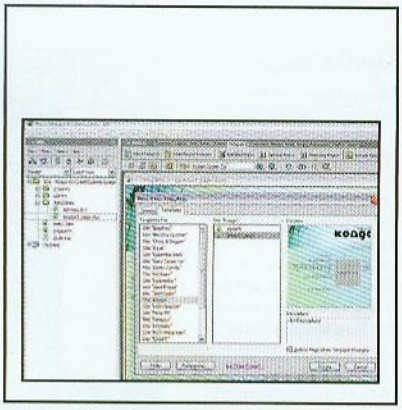
Nested regions

Creating editable regions within a nested template is a great way of restricting content amendments within pages, but they do have certain limitations. Because the nested regions rely on the parent template only, regions that have been designated as editable from there will be available for new regions to be created into. In other words, you can't create editable regions outside those specified in the parent file.

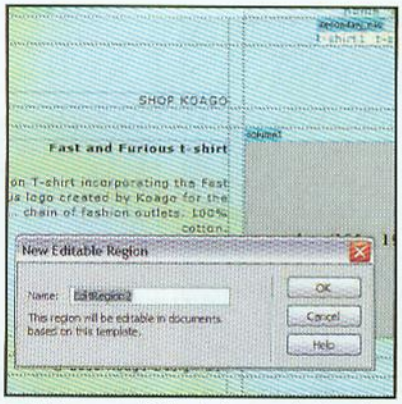


10 Next, add the content from the right-most column, ignoring the Add to Cart button, to complete the basic product page layout. Again, some basic formatting may be necessary to recreate the look required by the original *Photoshop* design.

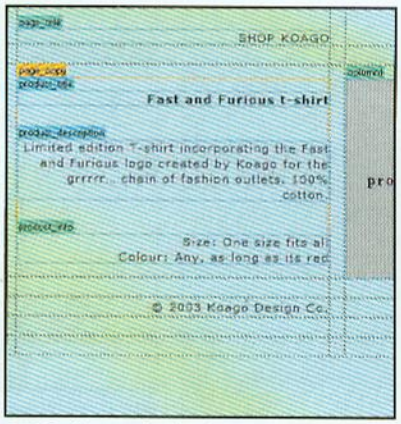
11 With our product layout complete we can now choose to save the file as a template via File>Save as Template. Provide a suitable name such as product_page and click the Save button to create the basis of a nested template.



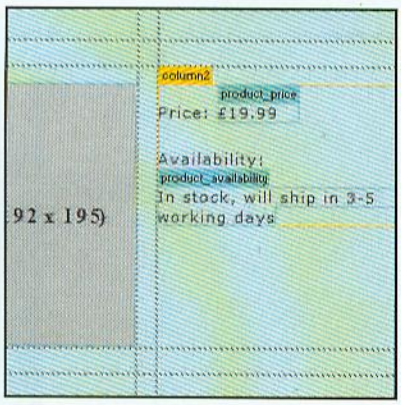
12 As when we created the original generic.dwt template, our new product_page.dwt file is now available to use and is stored in the dedicated Templates folder. You can now create new pages from this specific template by selecting File>New and choosing the required template.



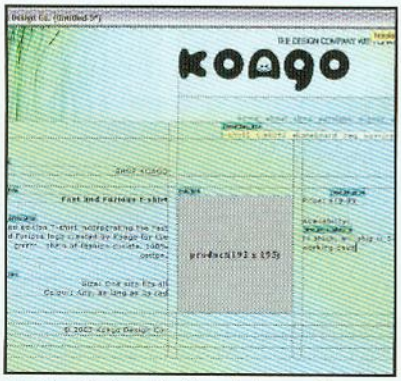
13 We can now further limit our product template by including editable regions for particulars relating to our product pages. Select the product title and click the Editable Region button found on the Templates tab of the Insert panel and name this: product_title. Then click OK.


SHOP KOAGO
Fast and Furious t-shirt
Limited edition T-shirt incorporating the Fast and Furious logo created by Koago for the grrrrr... chain of fashion outlets. 100% cotton
Size: One size fits all
Colour: Any, as long as its red
© 2003 Koago Design Co.

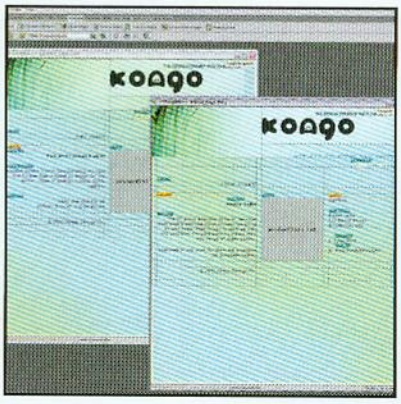
14 Do the same for the description, size and colour information, giving them suitable names. You'll then see how your nested regions form part of the original page_copy region determined by the parent template.



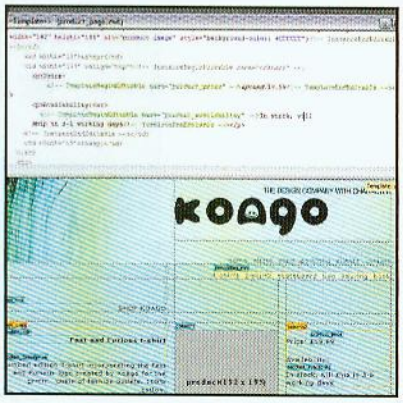
15 If the need arises, you can get much more specific than we have here. In the column2 cell you could just select the actual cost for conversion into an editable region. This would ensure that any formatting contained within the surrounding text remains undisturbed by future page amendments.



16 Save the template file again to commit your changes and create a new page using the product_page.dwt file as the chosen template – you'll be able to see the benefits of creating a nested template. Creating a new product page is now as straightforward as amending some key variable information.



17 You can continue to create nested templates for other areas of the site that may require duplicate content structures, such as the portfolio and client areas. Admittedly, our project only has a handful of examples, but doing this will ensure that future entries retain a consistent appearance.



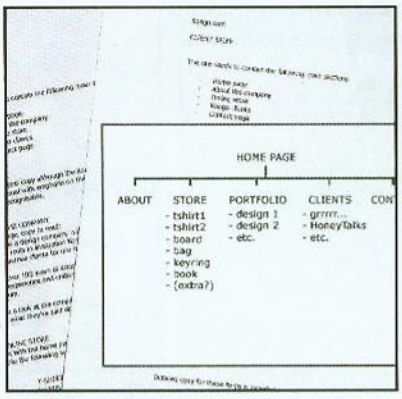
18 Finally, it's worth checking the template syntax again to address potential problems. One of the most common problems is the placement of a region outside a surrounding tag such as the paragraph <p>. If you're confident with the code, duck into code view to amend if necessary.

Part 3: Creating pages

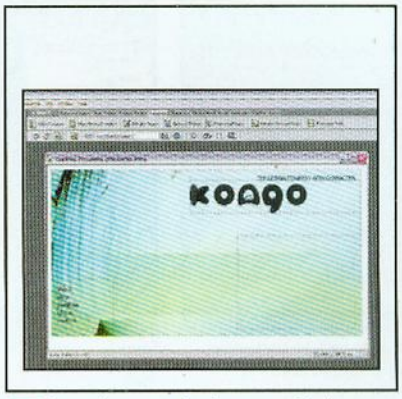
Use your templates to create the actual pages your site will deliver…

Files and folders

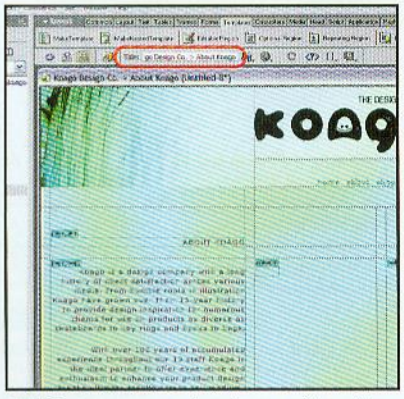
Organising your files within logically named folders helps to determine the address of a page. If a page is to act as the main page for such a folder, name it index.htm because this determines that the file will be the primary page within that directory. For example, if the main shop page is called index.htm within a shop directory then you can distribute the simple address of: www.koago.com/shop. Individual products contained within this directory need referencing from their unique file name such as: www.koago.com/shop/book.htm.



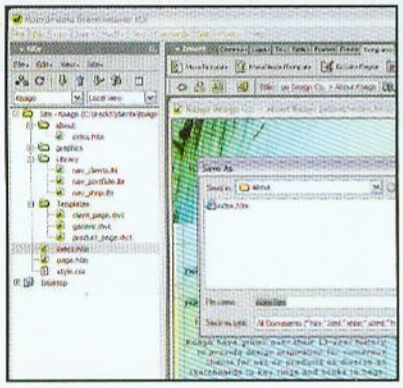
1 Before adding content, you need to look back at our original content.doc and the site/page structure we determined in the first section. This gives some indication as to the number of pages necessary as well as how they can be organised within our site.



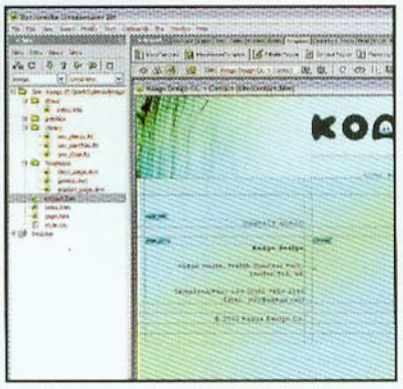
2 As our homepage is unique, we virtually completed that at an earlier stage. However, it's still waiting for the links to be activated once the subsequent pages are in place. A good starting point is to create another simple page like the one with the About information.



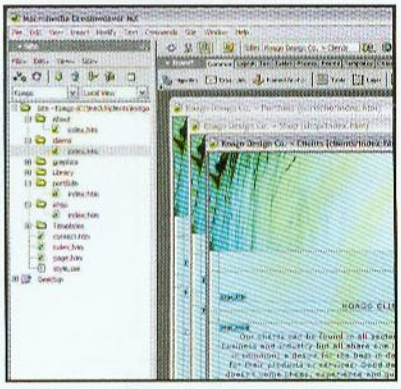
3 Create a new page from the generic template and copy the content from the content.doc file. Remember to change the title to suit the page purpose as well as updating the HTML page title within the document information at the head of the page.



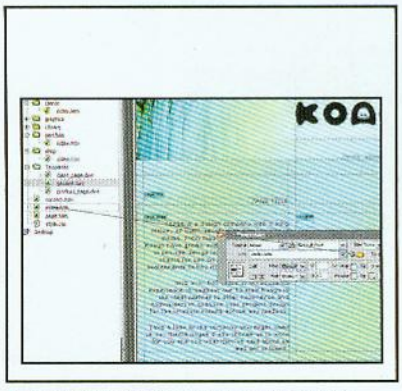
4 With the basic content in place, save the file as index.htm within a newly created directory named 'about'. The file name determines which file is the default file for this directory and provides the logical address of www.koago.com/about without any extended file names being necessary.



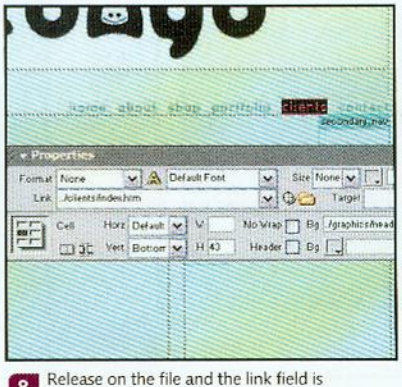
5 Close the file and create another new page from the generic template, this time using the contact information to populate the page. Save this page as contact.htm within the root of the site – this file doesn't need a folder creating as it's unlikely to need additional pages in the future.



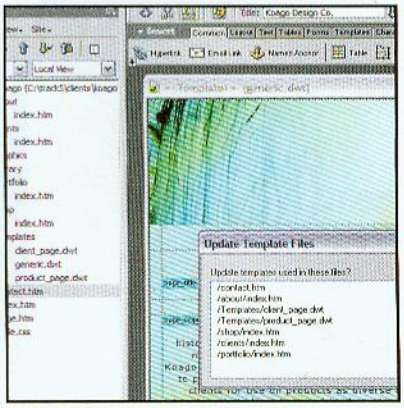
6 Repeat the process using the generic template for the shop, portfolio and client's intro pages saving these as index.htm in their respectively named folders. Don't worry about images at this point – just get the basic content in place and the files saved.



7 With the core pages created, we can now finalise the links to our primary navigation. Open the generic.dwt file from the Templates folder and select the 'home' link. By the link field is a crosshair icon – click and drag from here to the index.htm file in the root of your site.



8 Release on the file and the link field is automatically updated to reflect the relative link to that location. Repeat the process for the remaining primary navigation items, making sure to link them to the relevant index.htm file within their suitable folders (and the contact.htm file for the contact link).

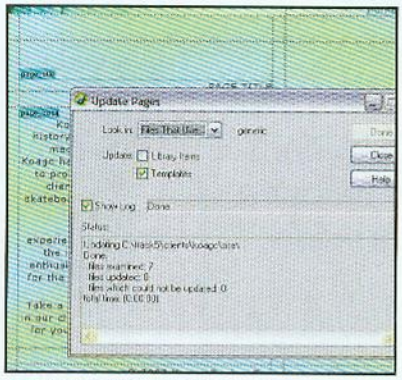


9 Save the amended template and *Dreamweaver* prompts you to update files based around this template. Notice the nested templates are included in the list so there's no need to update them separately. ›

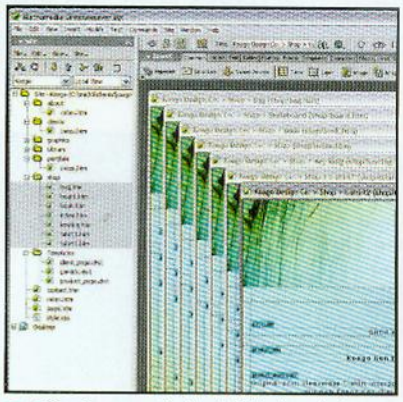
Creating pages continued...

Browsers and platforms

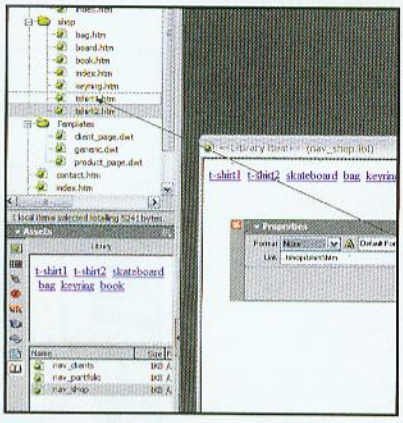
Keep a constant eye on how your pages look, not only in different versions of Internet Explorer but also Netscape and Opera. Macs and PCs have interesting quirks so you should test your pages on as many system combinations as possible. As you gain experience in Web design you'll start to realise which elements can be problematic, but there really is no better test than to spend some time actually viewing the pages.



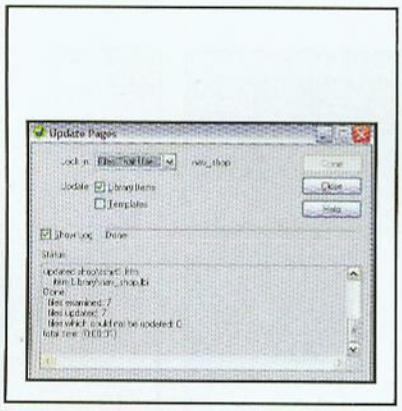
10 Click the Update button and *Dreamweaver* zips through the reliant pages updating the navigation accordingly. You're presented with a report on completion to inform you of the changes made. Ensure you read through the results to check that all the pages have been amended before closing the alert box.



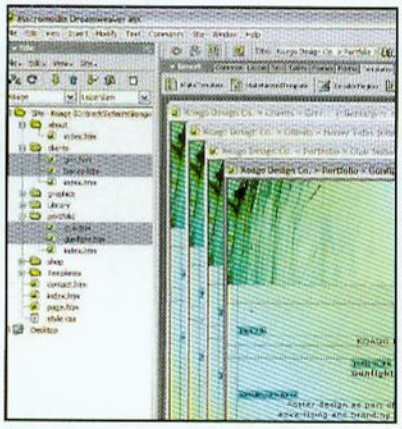
11 Creating pages for the individual product pages uses the same process as creating earlier pages, except this time you need to make use of the product_page template. Create new files from this and use copy from the content.doc, saving them with logical names in the shop folder.



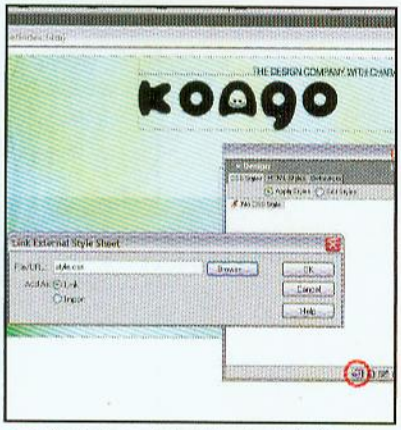
12 We can now amend the secondary shop navigation using the same techniques as we did for the primary navigation. This time, however, open the nav_shop Library item and change the links from there.



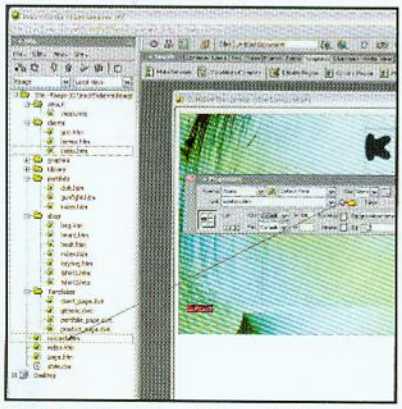
13 Saving the file prompts a similar update option as we experienced with the amended template, so click the Update button again for all files using the library item to be updated. Again, the same report is generated so check that before closing the alert box and the file.



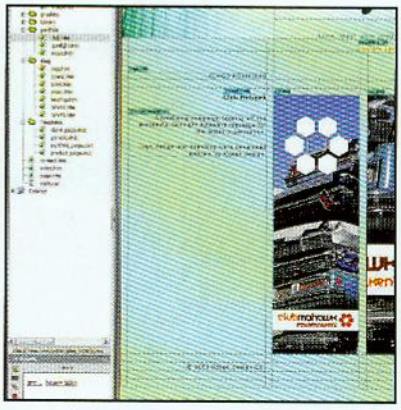
14 You now need to create similar pages and amend the secondary navigation library objects for the client and portfolio pages. Using the templates you've created makes this a very quick task and results in all your site pages being created.



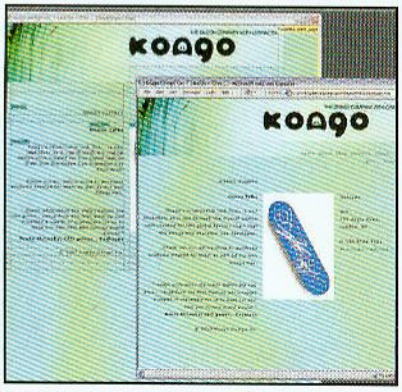
15 The final page to link into place is for the site homepage, so open the index.htm file on the root of the site. You'll notice, however, that this doesn't link to a style sheet, so create this using the CSS Styles panel and locating the style.css file.



16 Press OK and recreate the navigation as before by selecting each text instance and dragging the crosshairs onto the corresponding page in the site manager panel. Once complete, you can preview your site in a browser and all the pages should link together successfully.



17 With the fundamental site completed, you can now go through the pages adding suitable and specified images. Use *Photoshop's* Save for Web feature to optimise the images and don't worry too much about their height, but do keep rigidly inside the 192-pixel width limit.



18 You may also want to consider developing the text elements by creating new CSS styles to suit particular elements and making objects such as the client quotes more identifiable on the page. Have fun experimenting, but keep your changes consistent and keep checking how your pages look in the browser. CAP

SECTION **FOUR**

Extending the search

Dreamweaver offers a seemingly infinite amount of functions that can be installed and configured easily. These include complex features, such as our site-wide search engine…

One reason why *Dreamweaver* managed to revolutionise online content production is the application's extensibility support. Traditionally, creating complex features such as the Koago site search engine would have involved the services of a programmer, adding considerable time and cost constraints to a project.

Fortunately, we can now download dedicated extensions through the Macromedia Exchange site at [w] www.macromedia.com/exchange, which provides a centralised resource of extensions, from the simple Lorum Ipsum text generator through to more complex application server options. The majority of these extensions are free to download and provide designers with the technical ability to put some of the more complex tools in place.

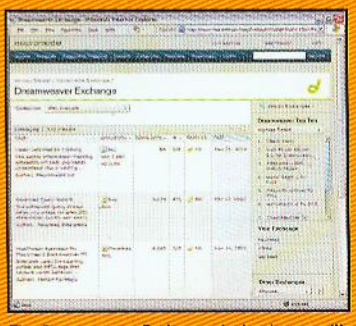
Since becoming established through *Dreamweaver*, the extensibility concept has now expanded over various Macromedia titles, including *Flash*, *Fireworks* and *ColdFusion*. And with new extensions becoming available on a virtually daily basis (created by existing users who have identified specific needs), there's likely to be still greater choice available in the future.

Managing extensions

The Macromedia *Extension Manager* is a small application that enables you to manage currently active extensions, making it possible to download and install multiple extensions without having to worry too much about interface overload.

From within *Dreamweaver*, the *Extension Manager* is accessed via the Help menu. It provides a simple interface that lists installed extensions along with a short description when selected. Of course, if you haven't used the feature before then the list will be empty, but clicking on the Macromedia icon will take you straight to the Exchange site, from where you can browse or search for suitable extensions for download.

Once installed, you can choose whether to make the object or behaviour available through *Dreamweaver* next time you launch the application. Simply check the On/Off field located to the left of your installed list.



The *Dreamweaver* Exchange site is where you'll find recognised extensions freely available for download. You'll need to register for a free account, but some advanced features are just waiting to be added to your site.

Illustration: Matthew Harvey
Photography: Richard Llewellyn

Creating extensions

Put something back into the *Dreamweaver* community by creating your own extensions for submission to the Dreamweaver Exchange site...

Although many extensions are written and provided by Macromedia, the vast majority are created by regular users, who have a specific requirement for their niche market that may be missing from the application.

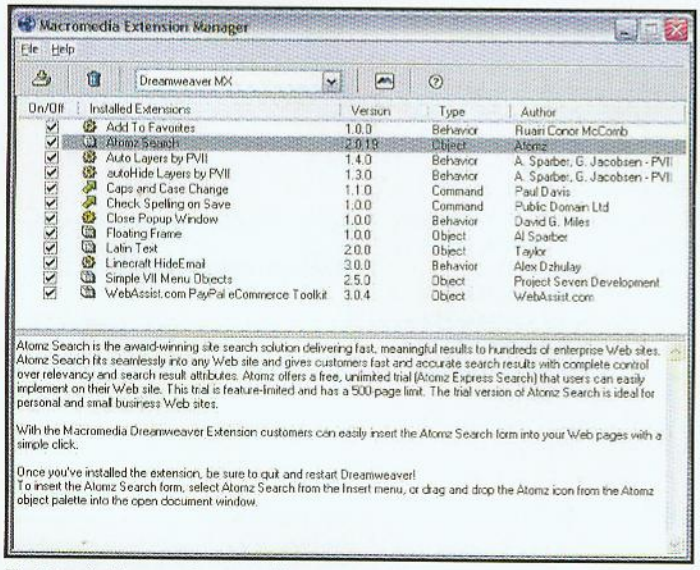
Creating and submitting new extensions is something that should be attempted only by users with more than average experience of not only *Dreamweaver* but also of the related Exchange site. There's no point creating a sub-standard extension for something if the definitive extension is already available.

Check out the Help menu for specific advice on creating extensions, including information on how to package them for submission as well as top advice on how to test your work before sending it in.

Take your pick

Running a search on the Exchange site often leads to various extensions offering a similar service. For simple functions, most people choose to download the most popular or most recent extension, but for more complex options, such as our search engine, a little more research may be necessary.

You may need to be aware of specific requirements, such as being restricted to a client or server OS or any technologies it may rely on – it's no good opting for a PHP Forum extension if your server will only support ASP, for example. Similarly, there's no need to confuse your site visitors with unnecessary functionality – choosing the most complex option may not be the most suitable for a basic task.


Macromedia Extension Manager
File Help
Dreamweaver MX
On/Off | Installed Extensions | Version | Type | Author
Add To Favorites 1.0.0 Behavior Ruairi Conor McComb
Atomz Search 2.0.18 Object Atomz
Auto Layers by PVII 1.4.0 Behavior A. Sparber, G. Jacobsen - PVII
autoHide Layers by PVII 1.3.0 Behavior A. Sparber, G. Jacobsen - PVII
Caps and Case Change 1.1.0 Command Paul Davis
Check Spelling on Save 1.0.0 Command Public Domain Ltd
Close Popup Window 1.0.0 Behavior David G. Miles
Floating Frame 1.0.0 Object Al Sparber
Latin Test 2.0.0 Object Taylor
Linecraft HideEmail 3.0.0 Behavior Alex Dzhulay
Simple VII Menu Objects 2.5.0 Object Project Seven Development
WebAssist.com PayPal eCommerce Toolkit 3.0.4 Object WebAssist.com
Atomz Search is the award-winning site search solution delivering fast, meaningful results to hundreds of enterprise Web sites. Atomz Search fits seamlessly into any Web site and gives customers fast and accurate search results with complete control over relevancy and search result attributes. Atomz offers a free, unlimited trial (Atomz Express Search) that users can easily implement on their Web site. This trial is feature-limited and has a 500-page limit. The trial version of Atomz Search is ideal for personal and small business Web sites.
With the Macromedia Dreamweaver Extension customers can easily insert the Atomz Search form into your Web pages with a simple click.
Once you've installed the extension, be sure to quit and restart Dreamweaver!
To insert the Atomz Search form, select Atomz Search from the Insert menu, or drag and drop the Atomz icon from the Atomz object palette into the open document window.

The *Extension Manager* software provides access to all your installed extensions through a central interface. Information about a selected extension is provided in the lower pane – disable those not required by clearing the tickbox in the On/Off column.

We opted for the Atomz Search tool for a number of reasons: the licence is freeware on sites with less than 500 pages, making it an ideal tool for personal or small business sites; and the title is well established, providing search options for various respected sites. If you need further persuasion, the tool has also won various awards including the 'Best of Show' award for Web Content Management at the AIIM 2003 Conference.

How it works

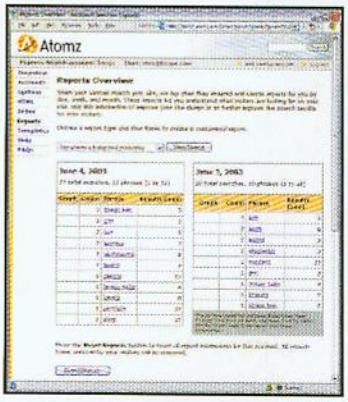
One of the main strengths behind the Atomz Search engine is its simplicity and effectiveness. Not only is it easy to configure, but most of the processing is carried out externally to your site. This means that there's no need to keep track of complex scripts or unrecognisable directory contents within your site.

Once an account is created, Atomz Search crawls through your site, starting with entry points that you've defined (as the site manager) and following links from page to page. The body content of the page is indexed along with the title and any meta tags that have been embedded, including those for keywords and descriptions. The resulting index is then stored remotely on the Atomz application network, waiting for retrieval through a client search.

Any pages that are to contain the search form need only a small amount of HTML to provide the function interface. This can be added manually if your coding skills are up to the task. Alternatively, it can be automated through the Atomz Search *Dreamweaver* extension, which we'll be looking at during the next tutorial within our developing Koago site.

Once in place, a site visitor can choose to run a simple keyword search through a form embedded into a regular page. Alternatively, they can perform more complex filtering by defining keywords to be searched between a time period or within specific areas of a page, such as the body or title or even within indexed URL or meta information.

The results are then fed back into an easily recognisable format, which allows users to preview potential listings that match their request before clicking through to what they consider are the most appropriate pages.



With an Atomz Search account configured and running on your site, you can access basic reports providing information on the keywords your site visitors use to locate content.

Reports and statistics

With a site configured and the pages indexed, Atomz provides a few bonus features that will be of interest to most site managers.

Templates are available to modify the appearance of the default page, or you can choose to create your own look and feel to further match your site. There are also various options available to manage your account, including providing control over language preferences or excluded words, if you prefer to remove common words or phrases to improve the quality of your search results. Similarly, if your site relies on frames, you can determine how the results interact with such a feature.

One of our favourite features, though, is the support for reports. This function logs user searches into a visual graph format, so that you can see exactly what your site visitors are searching for. This kind of return on the feature is crucial for discovering exactly what your site visitors are looking for and, when used in combination with your site's log reports, can offer a reasonably accurate insight into what draws people to your site.

However, this feature only works from an online site, so before we can demonstrate it, we need to upload our site in its current state. Go through the steps on the following pages to configure *Dreamweaver* with your hosting space. You'll need to adapt the remote server information to suit your particular situation, then subsequent parts will show you how to set up your Atomz account and integrate the search engine into your site.

You'll be amazed at how little technical information is necessary to embed such a complex element into your site, thanks to *Dreamweaver's* extensible options.

Files on disc
You'll find the components needed to complete this tutorial in the Tutorial\Section 4 folder on the CD.

Part 1: Configuring remote access and uploading the site

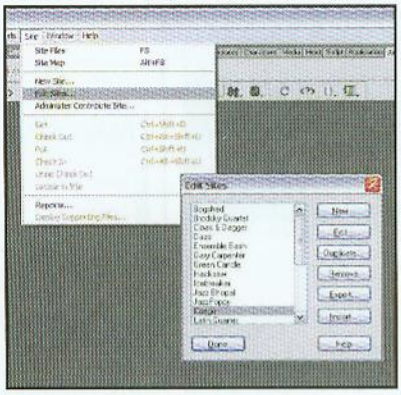
For the Atomz Search to work, it needs to index your site from the server. Therefore, we need to upload the site in its current state…

Remote access

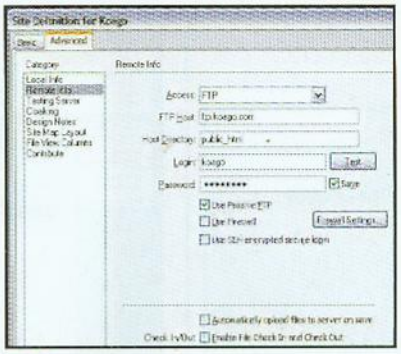
Obviously, we can't give all our readers access to the actual Koago server, but the following principle will work with your own hosting space as long as you submit your own FTP address and username/password details. You will need your site online for the search and shopping cart aspects of this tutorial, but signing up with one of the many free hosts such as Geocities will do the trick as you learn the ropes.

Passive FTP

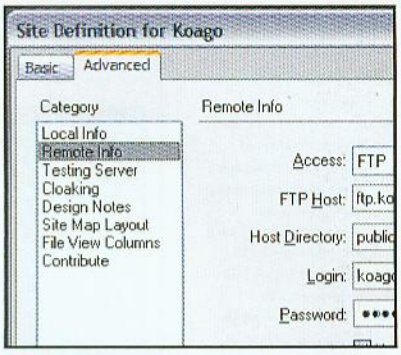
If you have a local firewall in place, you may need to opt for the Use Passive FTP settings or specify more detailed firewall settings. Passive FTP should get round most basic issues such as Norton Firewall (which may be fussy about what goes in and out of your computer). However, you'll need to get details from your network administrator if passive FTP isn't enough and the more corporate firewall settings need configuring.



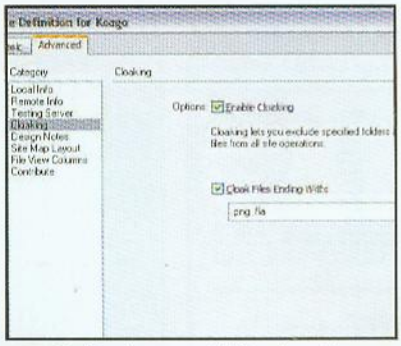
1 We configured our local site details earlier so that we could work in *Dreamweaver*. To enter our remote information, choose Site>Edit Sites, select the Koago site from the dialog box and hit the Edit button.



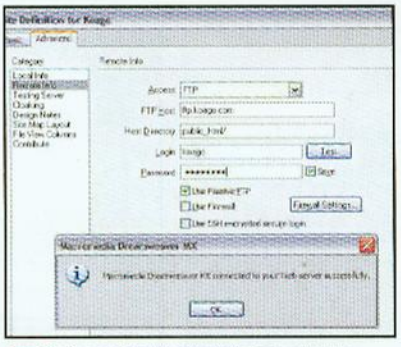
2 This prompts the Site Definition information, offering Local Info by default. Select the Remote Info category in the left-hand column and choose FTP access. Complete the information for the FTP host and the login/password details. You may need to enter a host directory depending on your server set-up.



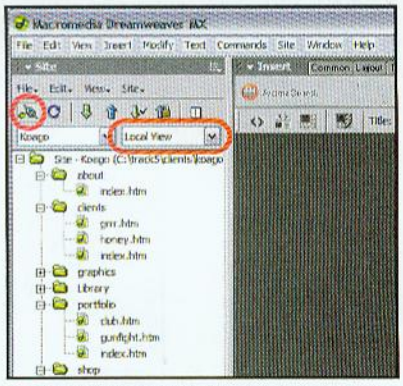
3 The remaining categories can be ignored for this project. The Testing Server is used with dynamic sites, and Design Notes are only really used in large collaborative sites. Categories for Site Map Layout and File View Columns provide preferential options, and Contribute is only available when the software is installed.



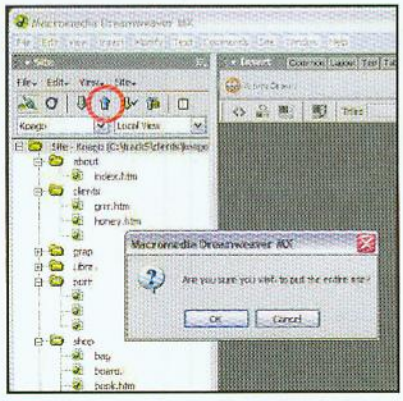
4 Cloaking, however, might be relevant for certain projects to define file types that should remain hidden within *Dreamweaver*, making them impossible to upload accidentally. This is particularly useful for original *Flash* .FLA files, which you might store within the same local directory as their related .SWF files.



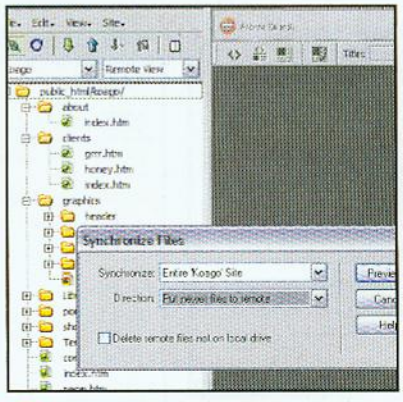
5 Before exiting the site definition, return to the Remote Info category and click the Test button to check your connection. If all's well, *Dreamweaver* will inform you of a successful connection; otherwise, check your details again and/or tick the Use Passive FTP option if you're behind a firewall.



6 Click OK to return to *Dreamweaver* and open the Site panel via Window>Site. Use the relevant icon at the head of the panel to connect to the remote host – once connected, the icon will turn green – and make sure you're in Local View.



7 To upload the entire site as it currently stands, simply select the top-most Site entry and click the upward-facing blue arrow to put the selected (all) files onto the server. An alert box appears, asking if you're sure you want to take this action. Click OK to continue.



8 The process of uploading the site may take a few minutes, depending on your connection speed. Once completed, the Remote View of the Site Panel should now display your duplicated files on the server. Use the Synchronise option under the Site menu to check they're all there.

9 Synchronise: Entire 'Koago' Site. In the Direction field: Put newer files to remote. Then click the Preview button for *Dreamweaver* to make local and remote comparisons. If the returned message is 'No synchronisation is necessary' then your site is up and running successfully.

Part 2: Installing the search extension

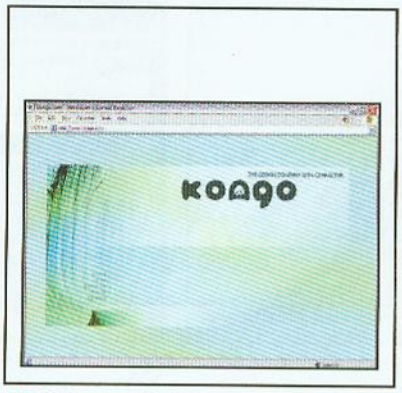
Install the required Atomz Search extension, register your account and index your site...

Managing extensions

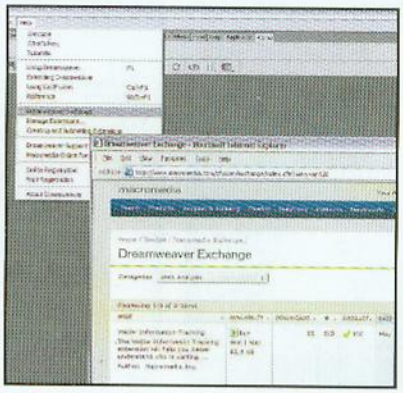
The *Extension Manager* is an incredibly simple piece of software. You may only use it for deciding whether to have an extension active or not, because most downloaded extensions will automatically prompt the application when required. If you have other MX titles installed, you might want to play around with other extensions by selecting an alternative software title from the drop-down menu.

Exchange registration

For full access to the Macromedia Exchange site, you'll need to register your interest. This is a free process and if you're going to be making much use of *Dreamweaver*, it's an essential step before you can extend your Web-authoring software. Once registered, you'll be able to download as many extensions as you feel necessary as well as having access to the other exchange sites for *Flash*, *Fireworks* and *Director*, among others.

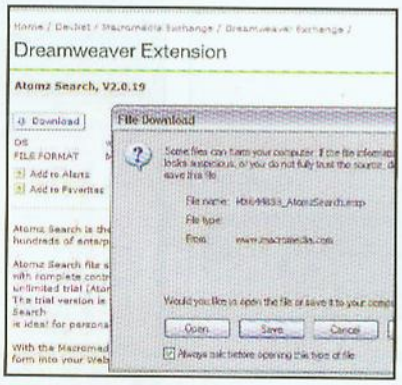


1 We've just uploaded the Koago site as it currently stands without the search engine or the shopping cart yet in place. It's the content that's important at this stage because this is what will be indexed by Atomz. Test that the site is available online at the domain.

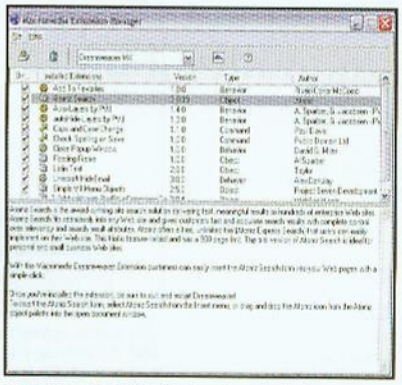


2 Before you can do anything with Atomz, you need to download and install the extension. Then you need to create and configure a free account with Atomz so they can locate and identify the site. Choose Help>Dreamweaver Exchange to visit the extension resource site.

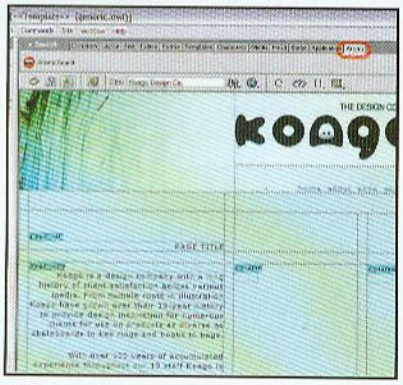
3 Use the link at the top of the right-hand column to Search Exchanges. Enter Atomz Search, choose the Dreamweaver Exchange and select your platform before hitting the Search button. You'll then be presented with a summary of your results.



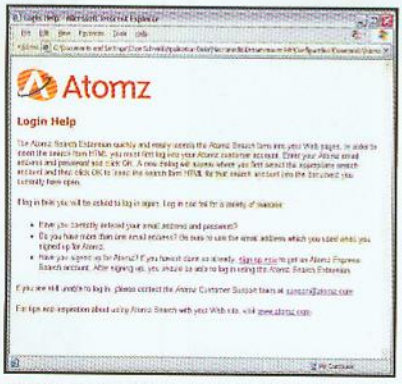
4 You can choose to view more information about a particular extension by clicking on its name, but as we're sure this is what's required, click on the Download link and choose to Open the file from its current location.



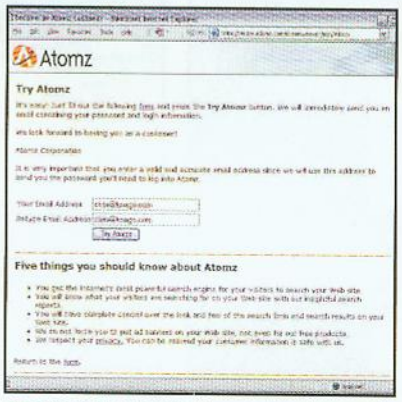
5 Once the 22KB file has downloaded, Macromedia *Extension Manager* will take over the installation process locally. Once the extension has been successfully installed, you'll need to restart *Dreamweaver* before it becomes available for use.



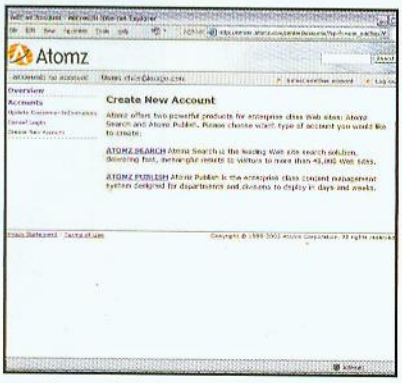
6 Open the generic.dwt file from within the Templates folder (or copy the site at this stage from our cover disc) and take a look at the Insert bar. You'll notice an Atomz tab has become available with the installation of the extension.



7 Attempting to insert an instance of the Atomz Search will prompt you for your Username and Password to access your account. However, because the account hasn't been set up yet, click the Help button to link to the appropriate page on the Atomz site to complete signing up for your account.



8 Submit your email address on the subsequent page, and details of the next stage will appear almost instantly in your inbox. Follow the link within the email to confirm your address and continue with the configuration of your account.



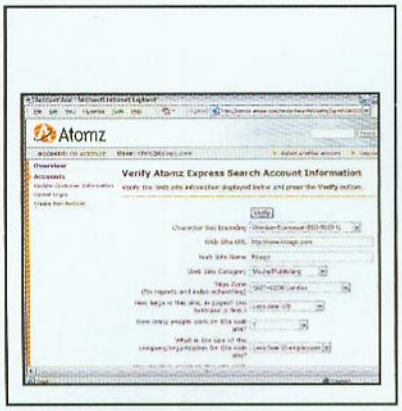
9 The email will direct you back to the Atomz site, from where you must follow a routine sequence of forms before you're given the option of creating a new account. Choose the Atomz Search option followed by the Express Search Account link to continue.

Choosing your extensions

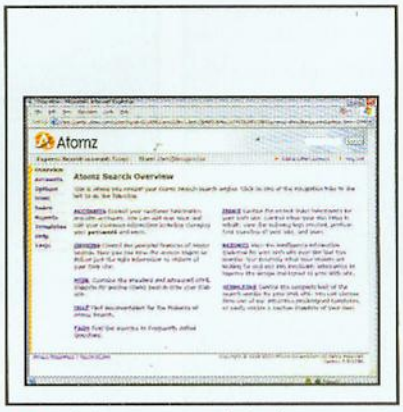
There'll be occasions when there seem to be various extensions that provide the same function. If in doubt, take a look at their ranking and/or the number of downloads a particular extension has had. This should go some way in helping you to decide whether one might be better than the other. You can also choose to click on the extension title for more information before committing to a download.

Multiple accounts

With your Atomz account configured, you can add multiple accounts that can be managed centrally via your log-in details, allowing site managers of various sites to index the contents of other sites easily. Once other sites have been configured, *Dreamweaver* gives you the option of inserting the relevant search fields into the correct site as a simple option.

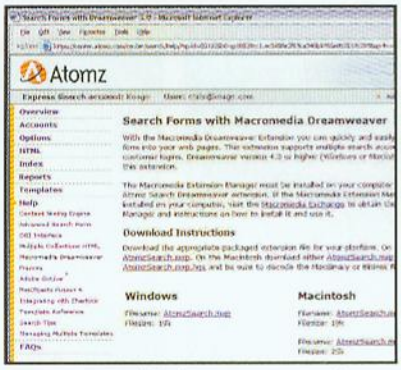


10 The next form enables you to create your search account. Complete the requested information carefully as this will be taken as the root info for your site. Click OK when done, verify your data and proceed into the main account interface.

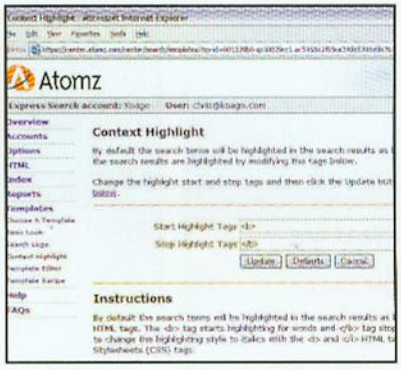


11 With the extension installed and your account configured, it's worth familiarising yourself with the online Atomz account management site. This is what you'll be using to tweak the engine to best suit your site.

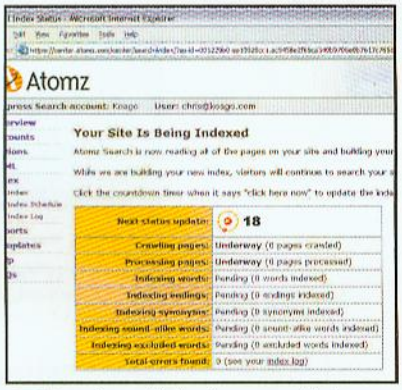
12 Existing accounts can be managed and new accounts created through the Accounts area. The Options area provides indexing, searching and crawling options for you to specify language and excluded words, any frame requirements a site may have, and URL entry points and masks.



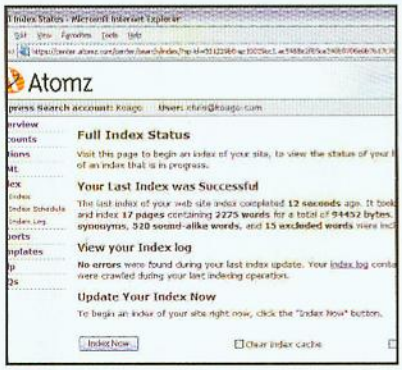
13 Having installed the Atomz extension, the HTML section is redundant – *Dreamweaver* will now look after the correct code insertion. The help and FAQ pages, however, will be invaluable to if you want to delve under the bonnet of the search engine in future work.



14 The Templates options are useful for modifying the appearance of your search results, which we'll look at later. The Reports options are particularly useful in the longer term because you can see what keywords and phrases site visitors are using to search your site.

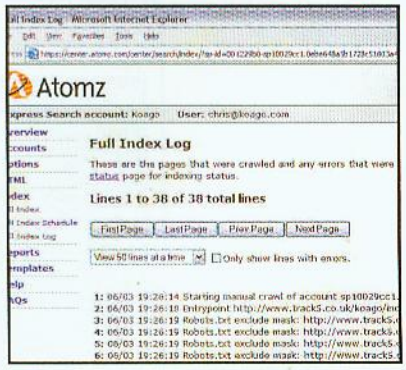


15 To continue with the site, you need to choose the Index options. Initially, you need to create a full index, so choose the Index option and click the Index Now button for Atomz to crawl through your site.



16 On completion, you're returned to the Full Index Status page from where a summary of your site is contained. It offers a useful summary of when the last index was completed along with the number of pages and words that Atomz has logged as your site content.

17 It's also worth scheduling when Atomz should automatically index your site in future. The Full Index Schedule link enables you to choose the day and time this should be carried out. For sites that are regularly updated, this is an important and time-saving feature.



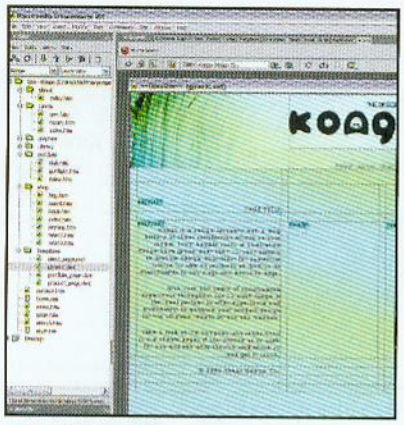
18 Finally, the Full Index Log provides details of pages crawled or masked from indexing. If you're unsure of the limitations of the last index, take a look here to see if any pages have been bypassed. If this is the case then such information cannot be returned in any search results.

Part 3: Search integration

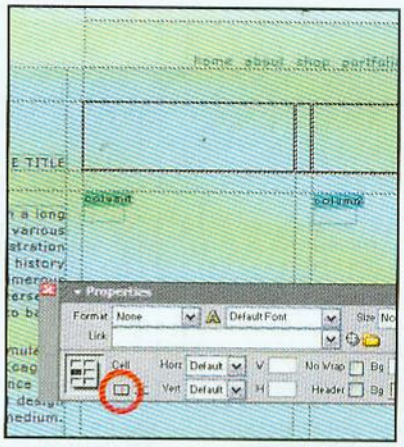
With the Atomz account configured and *Dreamweaver* extension installed, we can add the search form into our site…

Form alignment

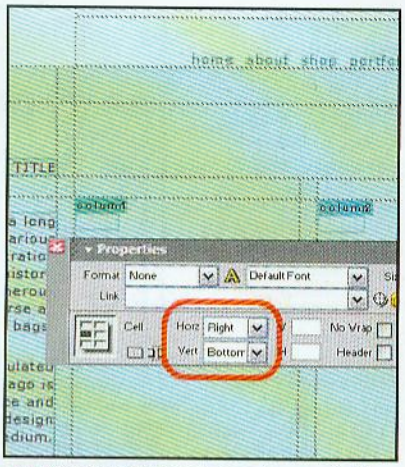
The insertion of a form can play curious tricks on a page layout, because the actual <form> tag that surrounds such an object automatically places space above and below in a similar way to the paragraph tag <p>. If this can't be integrated into your design, you can move the form tags further outside the code hierarchy. For example, moving the opening form tag inside the opening <body> tag and the corresponding closing tag </form> down to the bottom will alleviate the problem.



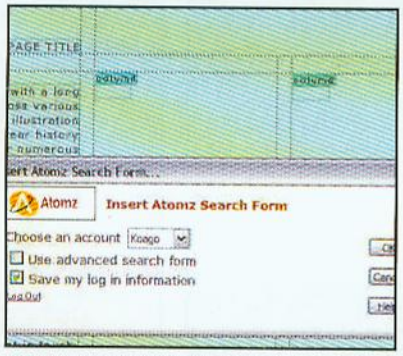
1 Returning to *Dreamweaver*, open the generic.dwt file from the Templates folder so that you can prepare the page for the basic search form. If you remember, this file acts as our site's primary template, with nested templates for the client, portfolio and product pages being reliant on it.



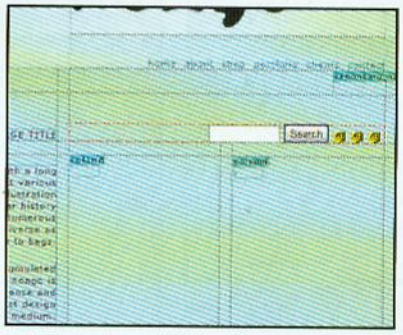
2 Our basic search field will be contained along the title row in the three cells above the column1 and column2 regions. Click and drag through these cells to select them simultaneously, then use the Properties panel to merge them together.



3 With the cells merged, place the cursor inside the new larger cell and align the contents to the right and bottom through the horizontal and vertical options found on the Properties panel.

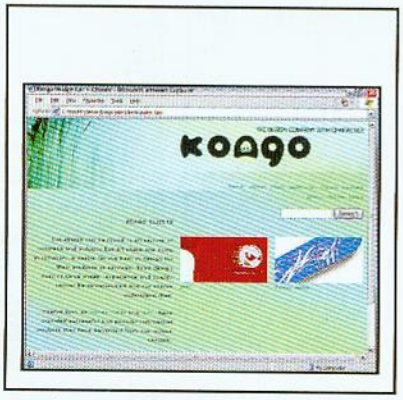


4 With the cursor flashing in the bottom-right of the cell, access the Atomz tab on the Insert bar and click on the Atomz Search option. Log into your account. Choose the Koago account and leave the default options as they are before clicking OK.

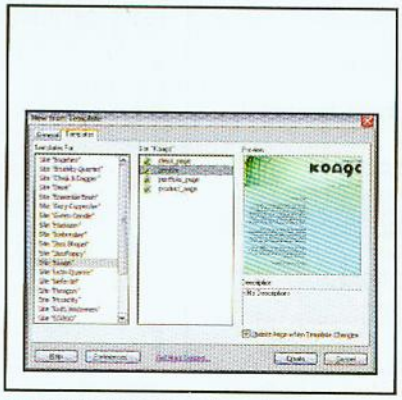


5 The basic search field and button are inserted along with some hidden fields. These should not be deleted, although we can remove the line-break between the search field and Submit button by placing the cursor between the two and hitting delete.

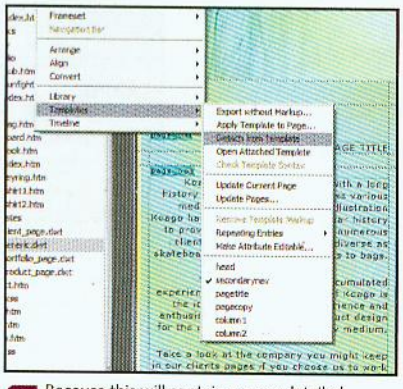
6 Before saving the file, choose File>Preview in Browser to test the layout and functionality of the search feature.



7 If everything's working as anticipated, save the template file in *Dreamweaver* to automatically update all reliant pages. These will include all the client, portfolio and product pages with the basic search form.



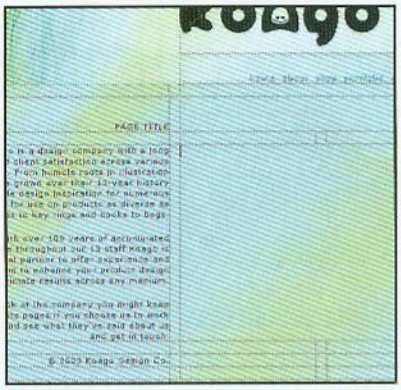
8 Atomz also enables us to create a more advanced search option, but this needs to be contained within its own page. Create a new page using the generic template and save this as search.htm in the root of the site.



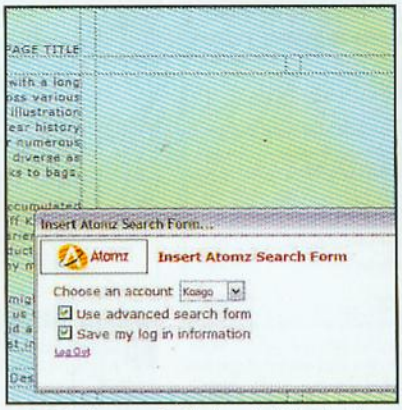
9 Because this will contain a more detailed search, the basic search option is redundant, and the two-column approach won't allow for the feature to be included. To convert the page into a unique layout, we'll need to lose the template via Modify>Templates>Detach from Template.

Checking templates

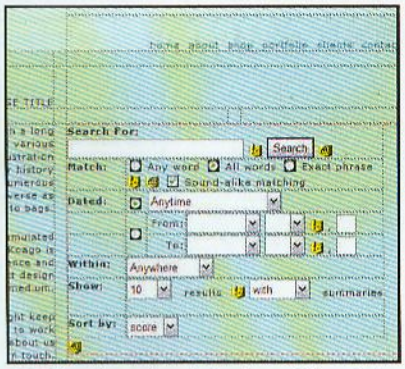
By their nature, templates affect multiple files when amended. If you're making changes to a template file, get into the habit of previewing any changes in the browser before saving the file. The browsers often have slightly different ideas of how content should be rendered, and relying on *Dreamweaver* to provide a constantly accurate representation of your work can be devastating. Previewing the page will highlight any potential issues before you commit them to any reliant pages.



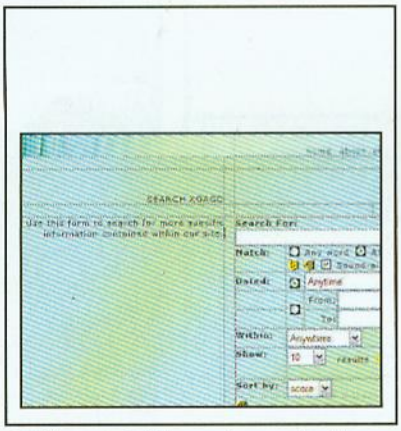
10 We're now free to delete the contents of the search cell and merge the cells that make up the empty body of the page to the right of the existing text. This creates the basic layout that will contain our more detailed search form.



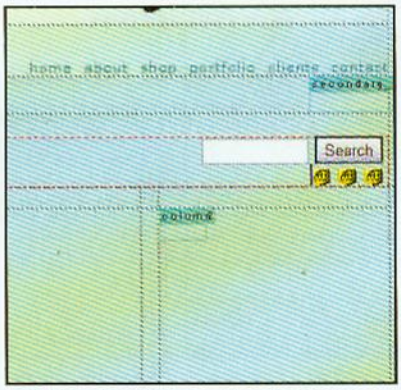
11 With the cursor in the large right-hand cell, select the Atomz Search option once more in the insert bar. This time, however, tick the Use Advanced Search Form option before clicking OK.



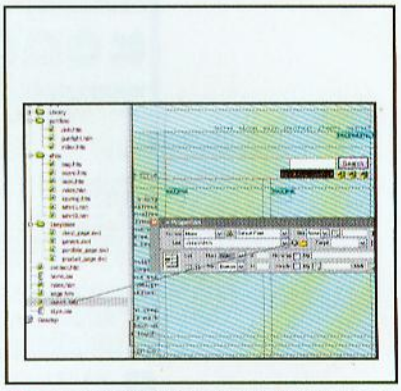
12 The advanced search form is now integrated into the page layout, providing much more detailed options. These include the provision to search for results that are date sensitive or can be found within specific page locations, along with result organisation preferences.



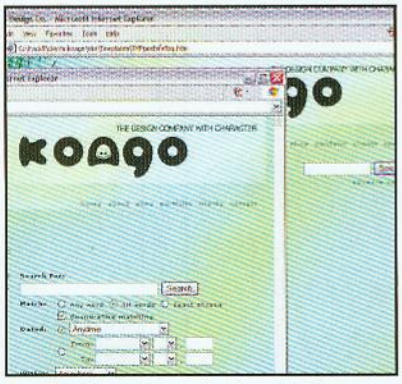
13 Now complete the page by amending the page and document titles, and revise the body text to be more suitable for the purpose of the page. Save and close the file.



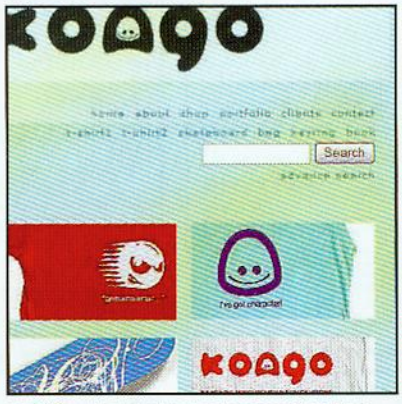
14 The only problem now is how to access the advanced search page from the remaining site pages. We could amend the navigation, but this might disturb our layout and create too busy an interface. Open the generic.dwt file once more and create a line break after the Search button.



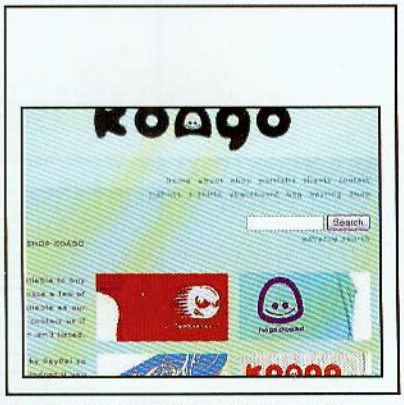
15 A more natural option is to simply include an 'Advanced Search' link under the default basic search form. Enter the text and create the link to the search.htm file by dragging the link field's crosshairs in the Properties panel to point to the corresponding file.



16 As with any changes to a template file, preview your amendments in the browser before committing changes to reliant files. Test the form again as well as the advanced search link – any problems caught now won't be passed onto other pages.



17 Once the template has been saved and has updated subsequent pages, take another look through the site. Notice that on pages containing secondary navigation, the search field is a little too close to the navigation.



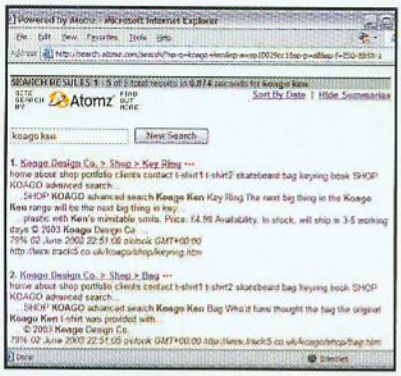
18 This is easily rectified by returning to the generic.dwt template file and inserting a break before the search form. Save the file to apply to your site and review the page – the navigation area should now be much tidier. ›

Part 4: Consistent search results

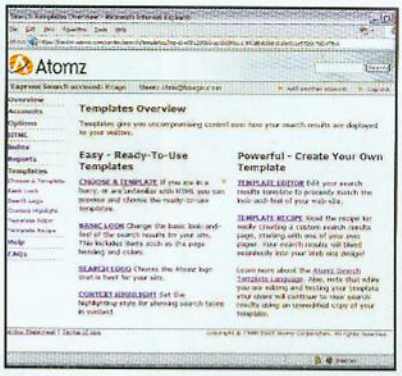
Modify the appearance of the Atomz search results to suit the Koago design…

Adding style

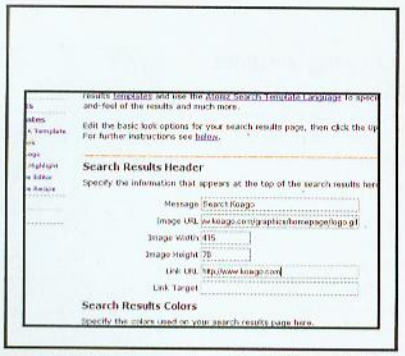
With a little coding experience, you could easily apply your uploaded style sheet document to the search results page by editing the code available through the Atomz site. Just remember that if you're linking to your style sheet, you'll need to provide an absolute URL to the file. This is because the search results are actually contained externally to your site.



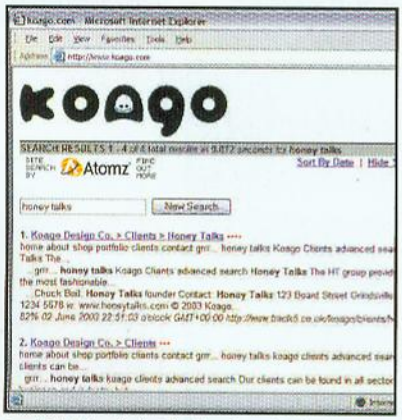
1 During the previous step, you may have noticed a difference between the appearance of our site pages and the one presented with the search results. Fortunately, we can adapt and improve the appearance to make them more consistent.



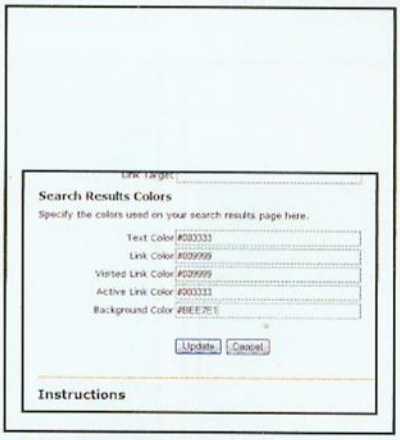
2 Log into your Atomz account online and click through to the Templates area. From here we could apply an available template, but this might still not be right for our purposes. Instead, click through to the Basic Look page.



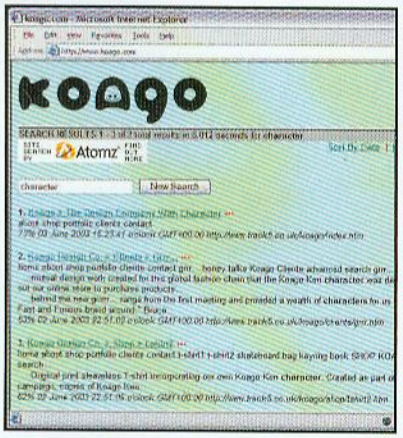
3 In the header area, type 'Search Koago' in the Message field. The remaining fields refer to an image we can embed. You can create your own, but we've referenced an existing image, entering its dimensions and pointing to the site homepage as the URL link.



4 Click the Update button found at the bottom of the form and test out your changes by running a search on the live site. Notice how the logo now appears at the head of the page, providing a link back to the homepage, while the alternative text reads 'Search Koago'.

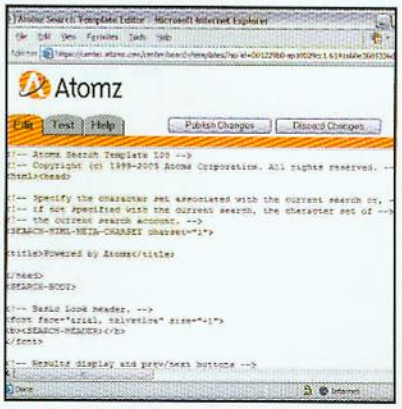


5 Return to editing the basic look of the page through the Atomz management site and scroll down to the Search Results Colours form. This information can be taken to mirror the style sheet specifications as we've done here.

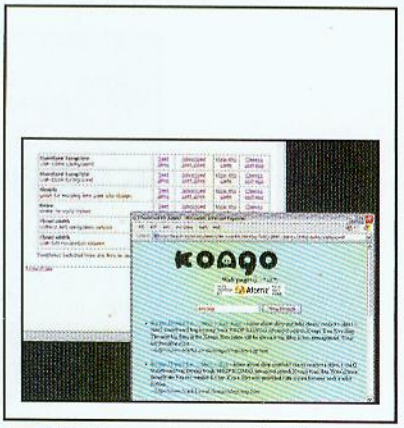


6 Again, click the Update button to save your preferences and test the file on the uploaded site. You'll notice we've managed to provide colour consistency between the site and the search results, leading to further integration.

7 As the code held on the Koago site that activates the search is actually quite simple, it's possible to further amend the code without disrupting the functionality of the search. However, you'll need a fair amount of coding experience to do this successfully.



8 Similarly, you can amend the code that controls the search results page through the Atomz site by going into the Templates area and clicking through to the Template Editor. This, too, requires fairly advanced coding skills and errors can be devastating – edit with care.



9 In the Choose a Template option, there are ready-to-use templates that will inherit your configured basic look. Use the Test Drive option for a preview of these easier options and, if one appeals, simply hit the Choose and Use link. ca-p

SECTION **FIVE**

Setting up a shopping cart

Use WebAssist's PayPal eCommerce Toolkit to create a secure online shopping facility...

As with our site's search engine, the shopping cart is another complex piece of programming which would be almost impossible for a regular Web designer to build from scratch. The functionality involved presents numerous problems for even the most experienced programmer, but when it comes down to security considerations, a shopping cart is not to be taken lightly.

An integral part of any shopping cart is to be able to collect and deliver customer credit card details securely. Without appropriate consideration of server security, you could find yourself in trouble if such details were intercepted and used illegally.

Fortunately, various *Dreamweaver* extensions exist to provide the core code necessary for some pretty complex and effective shopping carts. However, you'll still need to consider server security and go through the whole complicated route of applying for merchant bank accounts and paying not inconsiderable amounts for secure, or even dedicated, server access.

We've opted for the freeware WebAssist PayPal eCommerce Toolkit extension for our project because it bypasses many obstacles, allowing a fully-functional shopping cart to be put in place in a matter of minutes. Secure server issues are already addressed, allowing potential customers to shop online in confidence.

A wide range of selling preferences is available through the PayPal site once our cart is in place, including shipping costs and sales tax.

How it works

Rather like the Atomz Search engine we've already embedded into our site, the WebAssist shopping cart is embedded through a *Dreamweaver* extension. This process requires only a small amount of code to be placed onto your pages because most of the functionality is actually handled through a third party site.

Once the familiar Add to Cart and View Cart buttons are in place, your site users will be able to select items for purchase through the familiar online cart system. The cart itself is displayed through a secure connection in a new pop-up window that's managed by PayPal. Purchasers can then add multiple items, change quantity levels or remove items as they require before moving to the checkout to complete their transaction. At the checkout, credit card and delivery details are taken and submitted securely to the seller.

From the user's point of view everything is familiar, intuitive and

Illustration: Matthew Harvey
Photography: Richard Llewellyn

amazon.co.uk
SHOPPING BASKET
FREE Super Saver Delivery
Welcome Chris Schmidt! Your shopping basket can be seen below (if you're not Chris Schmidt, click here).
RECENTLY VIEWED ITEMS
Items In Your Shopping Basket
Qty.
Strategies for ECommerce Success - Paperback
Usually dispatched within 7 to 8 days
Our Price: £35.94
Animation (Short Cuts) - Paperback
Usually dispatched within 2 to 3 days
Our Price: £12.99
Subtotal: £48.93
Items Saved For Later (Items remain for 90 days)

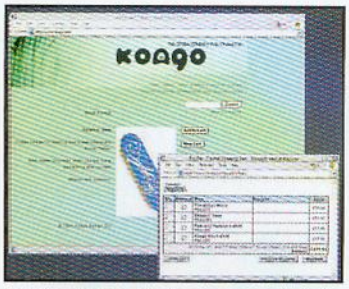
Creating a complex shopping cart such as that provided by Amazon takes time and money – merchant accounts, secure servers and extensive programming are required.

secure — the shopping cart window offers all the expected options for amending their order and PayPal even remembers selections if the cart window is closed. Crucially, the customer is confident of the security of their transaction because the cart is held on a secure server. This is made obvious by the padlock symbol in the browser status bar, along with the https URL prefix.

Making money

Another plus point with the WebAssist PayPal model is that the extension is free to use and there are no ongoing costs once the service is included on your site. However, rather than being a charity organisation, PayPal has found a clever way of generating revenue from the service.

Any money that's paid towards your shopping cart items is transferred onto your registered PayPal account, so we'll need to set up one of these before we can continue developing our site. From these received funds, PayPal takes a small percentage commission of between 2.7–3.4% depending on the



Using this WebAssist extension, you can include familiar e-commerce features that make use of PayPal's established security provisions.

type of PayPal account that the customer holds.

You'll also pay 50 pence to transfer funds to your regular bank account, a process that takes around 5–7 working days, which may prompt you to leave your money on your PayPal account until a reasonable sum has amassed. It must be assumed that PayPal makes interest on the money in your account, but given the simplicity of the WebAssist tool and lack of ongoing regular costs, these are reasonable terms of using the tool.

PayPal

Although WebAssist is the company that has created the extension software to integrate its shopping cart into your site, the actual transferral of money is all managed through PayPal.

PayPal is an established online payment company that found success through online auction sites such as eBay. It provides quick and cost-effective methods of transferring money instantly between people that may be located at opposite ends of the globe. By speeding up the payment aspect of a transaction, many people started turning towards this method of payment in preference to the slower method of sending cheques or money orders.

Its claimed user base of over 25 million customers in 38 countries is due to expand, with the weight of eBay behind it. This leading global auction site is now promoting PayPal as its preferred payment method.

Pros and cons

The integration of the PayPal method is a reasonable option for the small business. But many organisations making their primary income from online sales are probably better advised to spend more time and money, at least initially, on setting up their own secure servers and merchant accounts. There are still plenty of costs involved, but any received funds will go straight into the company's bank account, where they'll earn interest and be instantly available for them to make use of.

It can also take some time to get a PayPal account set up and recognised. And for good reason. If you're planning on sending or receiving large sums of money, PayPal wants to be sure that your account is active to prevent instances of fraud or deception. So although the model is ideal for our simple Koago store, you need to bear in mind that the registration process can be time-consuming — a definite disadvantage if you need your shopping cart up and running in a weekend.

Despite the lengthy registration process, there's one crucial advantage with the PayPal model as far as a customer is concerned; they do not need to have a registered PayPal account to make a transaction. The system allows anyone with a valid credit (or debit) card to select items for purchase and complete the transaction without putting them to the bother and general time-wasting of creating a whole new account.

Continue with our project and you'll soon see how easy it is to set up a secure online sales option for your site.

Files on disc
The files needed for this section of the project can be found in the Section 5 folder on the cover CD.

Direct sales

Cut out the PayPal middleman and get hold of your hard-earned cash quicker...

Our PayPal cart is fine for the smaller business, but it might not be practical for larger businesses who need money in their accounts as soon as possible for future investments, stock purchase or those lucrative director dividends...

The core elements you'll need to enable you to accept online payments independently are a) an Internet Merchant Account (IMA), which can be obtained from various acquiring banks, and b) some kind of payment processing system to act as your payment gateway, which is the kind of service offered by established companies such as SecPay or WorldPay.

Your bank should be able to put you on the right track for getting started. They'll provide you with IMA advice along with any preferred payment gateways they may have partnerships with. Have a chat with your bank manager and you'll soon be selling online.

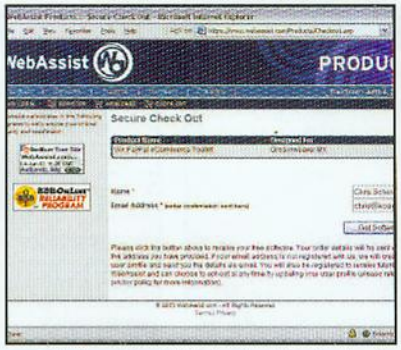
Part 1: Creating a PayPal account

The WebAssist extension and PayPal account are necessary for you to integrate your shopping cart and start receiving payments through your site…

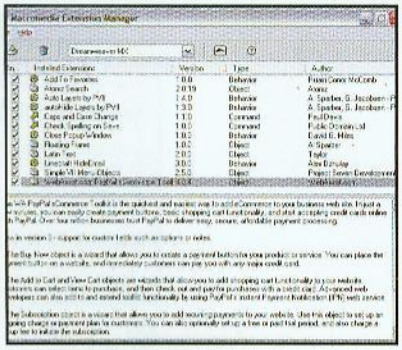
Plan in advance

Establishing a PayPal account takes time because your email and bank accounts have to be validated. PayPal makes a small charge from your chosen account, which is refunded into your PayPal account once the process is completed. Your bank statement for this will contain a reference code that must be submitted to PayPal to finalise the process, so using online banking to check your statement will speed up the process.

1 As with the Atomz Search engine, you can locate the necessary WebAssist PayPal eCommerce Toolkit through the Macromedia Exchange site. Once located, clicking the Download button takes you straight to the WebAssist.com Website, from where the actual download must take place.

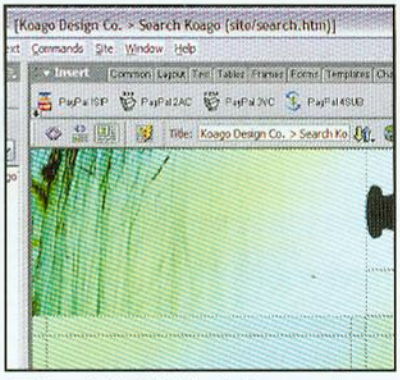


2 You'll need to add the free toolkit extension to your WebAssist shopping cart and then check out of the shop to accept the software licence agreement. You must submit your email address before you can actually initiate the download of the .MXP extension file.



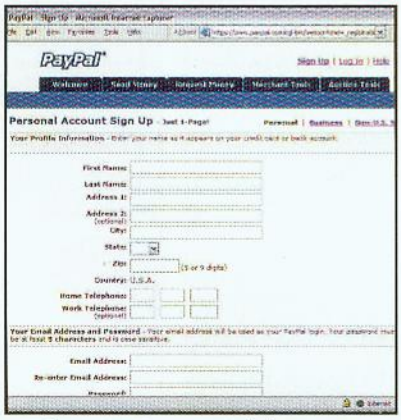
3 Choose the Open option in preference to Save, and the Macromedia *Extension Manager* will take over the installation of the file. Read through the associated information for the latest version updates, while you're restarting *Dreamweaver* to make the extension accessible.

4 As we'll be applying this to our product pages, open the first product, bag.htm, from the cover CD and take a look at the Insert bar. You'll notice a PayPal tab has been created. Click on this and you'll see a few options for adding shopping cart components to your site.

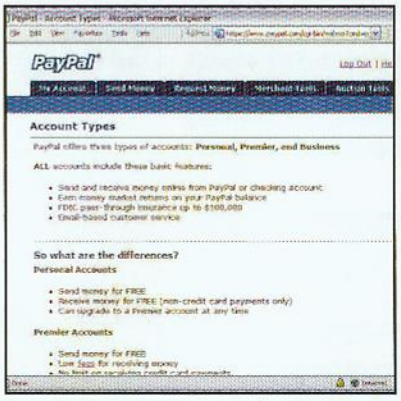


5 From left to right, these buttons comprise insertion buttons for Buy Now, Add to Cart, View Cart and Subscriptions. The Buy Now option is for single item instant selection, while the Subscriptions option enables you to accept regular payments such as subscriptions.

6 If you already hold a PayPal account, you can skip to the next part of this tutorial, otherwise you'll need to head over to **[w]** www.paypal.com to sign up for a new account. Click through the obvious central link to continue with this.



7 The registration process is fairly straightforward, although it can take some time for your account to become active. This is due to the security issues that need addressing when dealing with financial transactions.



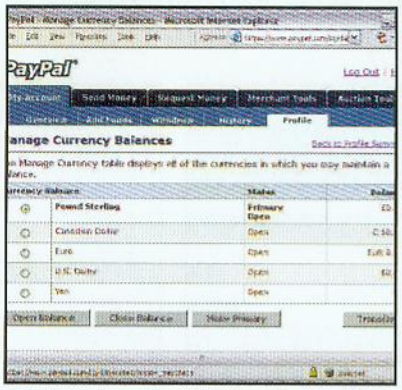
8 Once your account is active, there's some configuration necessary for our site. First, upgrade your account from the default Personal to a Premier account. This is a free upgrade, but you should take a look at the Accounts page for a summary of the benefits.

9 Most of the configuration necessary can be found under the Profile tab within the main My Account section. Account Information manages details such as your address. Financial Information tracks your bank and credit card details. Finally, Selling Preferences enables configuration of VAT and shipping costs and so on.

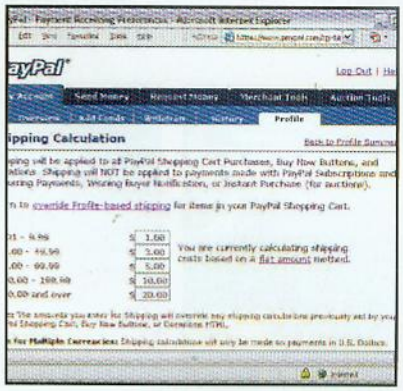
Creating a PayPal account continued...

PayPal shops

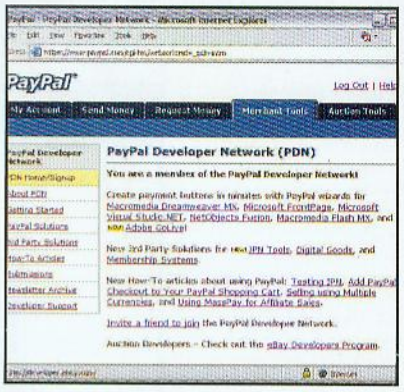
Although it isn't a requirement for your customers to have a PayPal account, some PayPal users do leave small amounts of money in their accounts and these are the people who are more likely to make spontaneous purchases. Registering your site with the PayPal Shops directory via the shops link in the bottom navigation will improve your site's chances of being visited by people with money to burn.



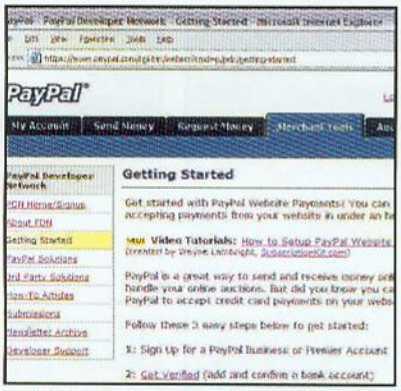
10 Account and Financial Information should be up to date from registration, but, if necessary, you can amend or add information or include alternate bank or credit card accounts for primary use. If you're anticipating sales from overseas, consider opening balances to manage relevant currency accounts.



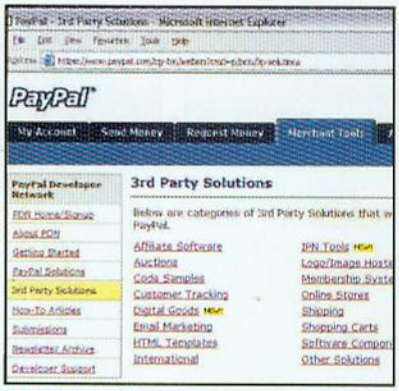
11 Selling Preferences contain a few elements for controlling the way our Koago store manages sales. Regional Sales Tax is a US-based tax system, but there's provision to override this for UK VAT. Shipping Calculation provides a method of determining how much postage is added to completed sales.



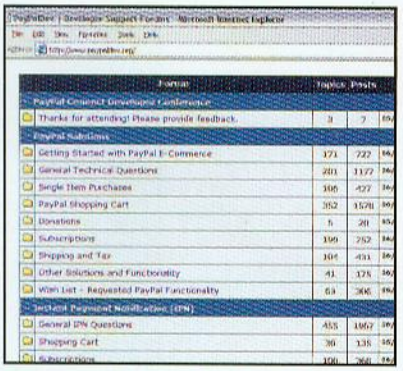
12 The main Merchant Tools tabbed area contains a number of features that help create elements for your store, but most of these are catered for through the WebAssist extension we've already installed. Instead, follow the link at the bottom of the page marked Developers through to the PayPal Developer Network.



13 This is a particularly useful area of the site for site managers to learn more advanced features for customising the functionality and appearance of their shopping cart. The Getting Started area, for example, contains various video tutorials and advice on figuring out how to set things up.

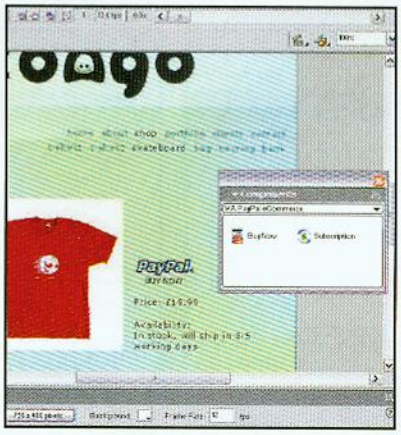


14 Similarly, the PayPal Solutions and 3rd Party Solutions areas provide invaluable advice on some more specific considerations you may come across as your store develops. These areas allow you to make your shopping cart functionality as appropriate as possible to your needs.

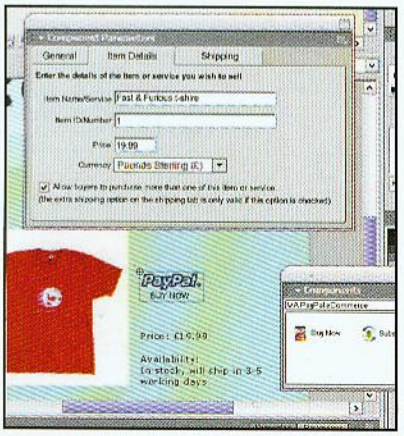


15 For more advanced topics, such as figuring out how to dynamically modify item costs, and descriptions how to incorporate streaming content through the VitalStream MediaConsole, you can always turn to the How-To Articles or consult the developer message boards at **[w]** www.paypaldev.org.

16 In addition to *Dreamweaver*, the PayPal model can also be integrated through *GoLive*, *FrontPage* and *NetObject Fusion* extensions as well as Macromedia *Flash*. The *Flash* extension has also been created by WebAssist and provides an easy way to embed similar features into your SWF files.



17 The *Flash* method is incredibly simple, installing a selection of dedicated components to offer the functionality of the shopping cart within the animated environment. A required component is simply dragged and positioned onto the canvas.



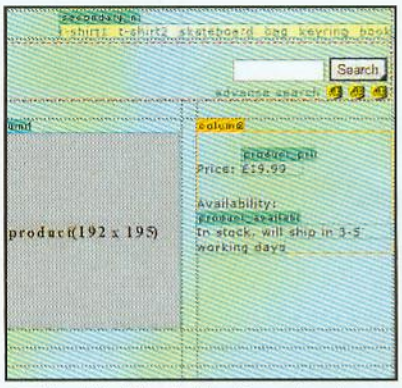
18 And once in place, the component parameters can be readily adjusted to enter your PayPal account details – along with information about the product and variable shipping costs – before being published and contained within the final .SWF file.

Part 2: Integrating the shopping cart

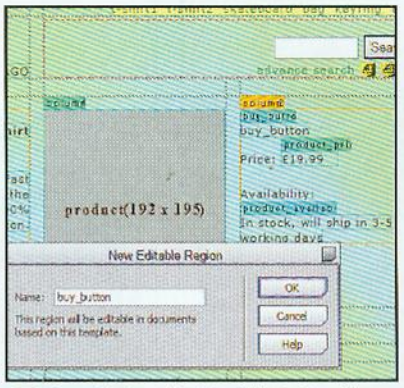
Prepare the product pages for the WebAssist PayPal tools and add online sales options to the site…

Fussy template regions

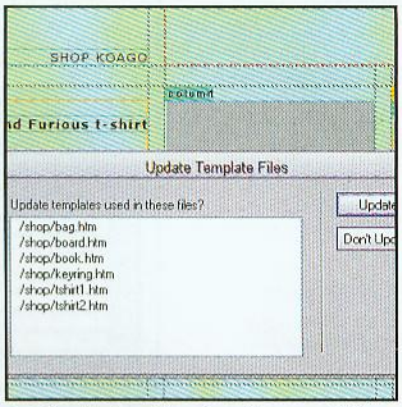
When inserting certain objects into a template region, such as our Add to Cart button, you may experience problems with alert boxes informing you that the inclusion of such an object would insert code outside the defined region. If this is the case, go back to your template and check you've included paragraph tags inside the editable region. Use the Insert>Tag option if you're nervous about editing raw HTML.



1 To add individual buttons for our shopping cart, we're going to need to adapt the product_page.dwt template file to provide an area where an Add to Cart button can be placed. Open the file and create a new line above the Price element in Column2.

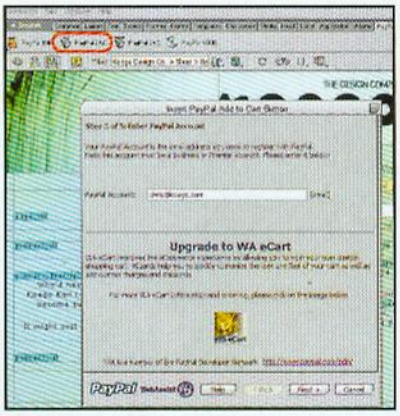


2 Access the template tab within the Insert bar and create a new editable region, calling it buy_button. Your text title will populate the empty entry. Delete the text because this will be replaced on each of the reliant pages, and insert a non-breaking space instead.

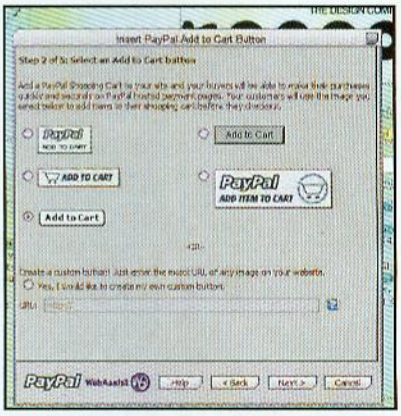


3 Save the template and you'll be prompted to update any files that rely on the product_page template. Make sure all your product pages are included in the list, and click Update to apply the change to the recognised pages.

4 Close the template file and open the first of your product pages, bag.htm, contained within the shop folder. You'll see how the file has inherited the new region, allowing us to add our WebAssist PayPal button directly onto the page.

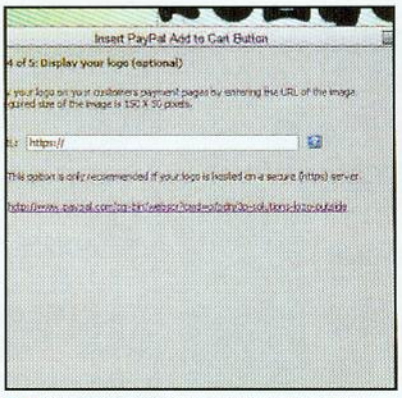


5 Place your cursor within this new region and select the PayPal tab on the Insert bar. Click the button marked 'PayPal 2AC' to prompt the button's configuration wizard. Enter your registered PayPal email address in the available field and click Next.

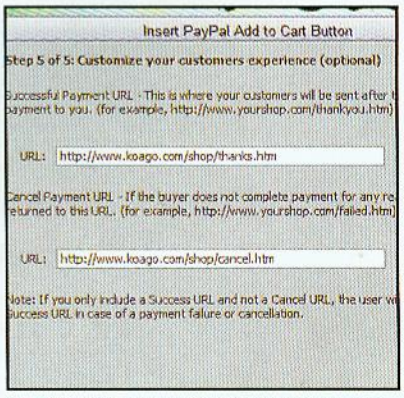


6 The following screen enables you to choose from a selection of Add to Cart buttons. Make your choice or create a custom button, and enter the hosted location of the file to personalise your site further. Click Next again to continue.

7 The next screen requires specific details for the item that's being offered. The fields are pretty self-explanatory, so go through them using the existing page content as a guideline. Click Next again to move on.



8 You're now given the option to display the Koago logo on your customer's payment pages by entering the absolute URL of the graphic file. As this is only recommended through a secure server, we've opted to leave this blank. Continue with the configuration by clicking the Next button.


Insert PayPal Add to Cart Button
Step 5 of 5: Customize your customers experience (optional)
URL: http://www.koago.com/shop/thanks.htm
URL: http://www.koago.com/shop/cancel.htm

9 The final screen of the configuration process responds to actions. It either bounces your customers through to pages that inform them of a successful transaction, or reassures them that their payment has been cancelled. Copy the links as shown above and we'll create these pages later. ›

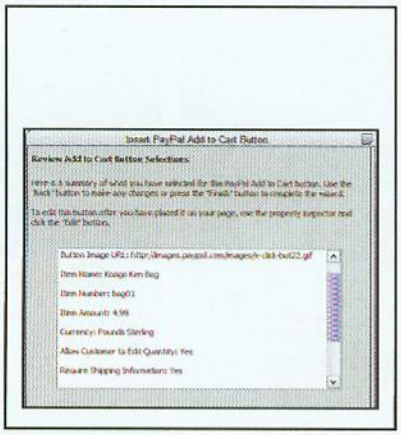
Integrating the shopping cart continued...

Testing your buttons

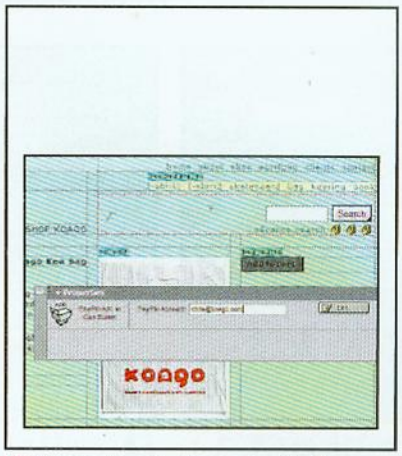
Although we've recommended making constant use of *Dreamweaver's* preview options for testing your pages, the temporary files this creates won't allow you to test how your buttons work. To do this, you'll either need to save the file and view the actual page locally, or upload to the server and test from there.

Cut and paste

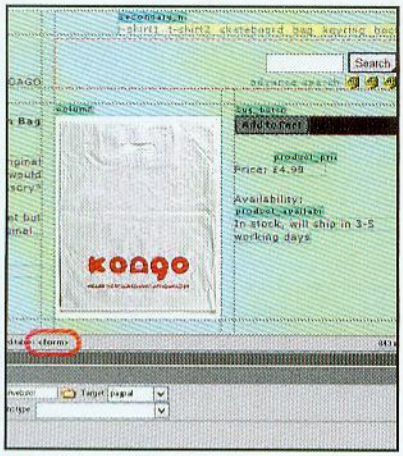
To save time when creating buttons for the other product pages, you could simply cut and paste the buttons from the original file. However, you'll need to make sure that you correctly amend the details or your store will be selling the same item from each page regardless of what the page states.



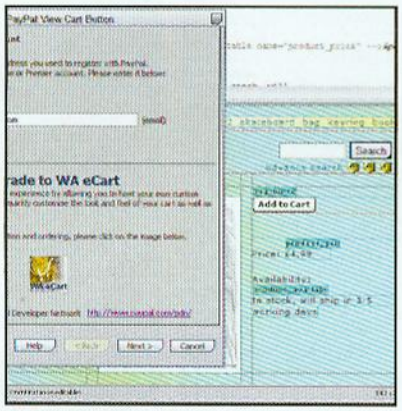
10 Clicking the Next button presents a summary of your configuration. Read through this carefully to ensure that all the information is correct because this will be passed on when the product is added to the buyer's cart. Use the Back button to correct entries, or click Finish.



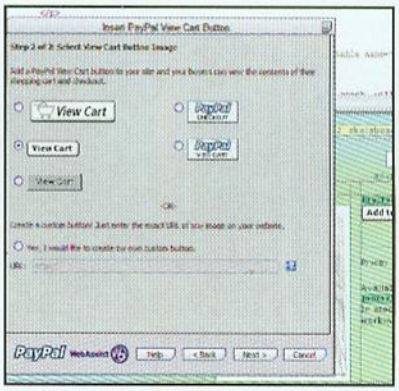
11 The configured button is now added to your page layout. Should you need to edit the button configuration at a later date, simply select the button on the page layout and use the Edit button to make your required changes.



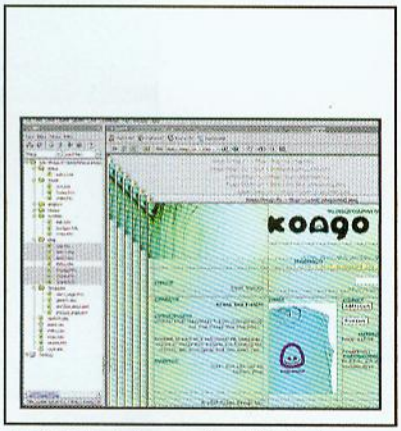
12 Adding the View Cart button uses similar principles, but careful consideration needs to be made to place this outside the existing button form. Click the button on your layout and select the <form> tag at the bottom of the page window.



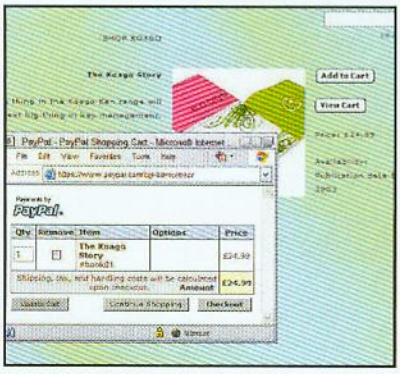
13 Use the right arrow key to escape the form and hit Shift+Return to create an HTML break tag <br> after your form. Then click the View Cart button marked 'PayPal 3VC' (found on the PayPal tab of the Insert bar) to prompt this button's configuration wizard.



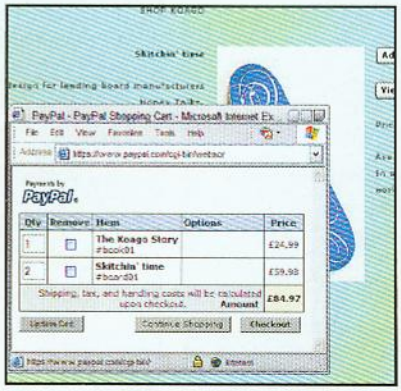
14 Your registered PayPal account email address should have automatically populated the first step, so click Next to choose the image to be used as the View Cart button. The option to review your configuration follows; assuming the basic information is correct, click Finish to position the button.



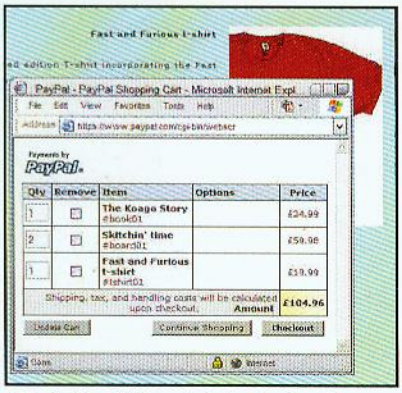
15 With the buttons configured for the Koago Ken Bag, you'll need to repeat the process for the remaining product pages, providing each item with its own unique configuration to suit the price and product description.



16 With our cart now in place, take a look through the site and test the buttons. Clicking an Add to Cart button should now generate a new window containing your cart items. You should see all the features you'd expect, including quantity updating and item removal options.



17 Your site visitors can view other product pages and, should they choose to add additional items to their list, the cart automatically updates itself. The cart will be brought back to the front so you can see its revised status.



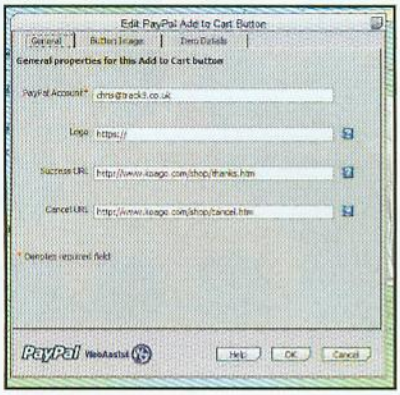
18 And should the cart window be closed accidentally, you can rest assured that your visitors' lists will remain intact. They can be retrieved instantly through the View Cart button so the visitor can continue with their transaction.

Part 3: Confirming transactions

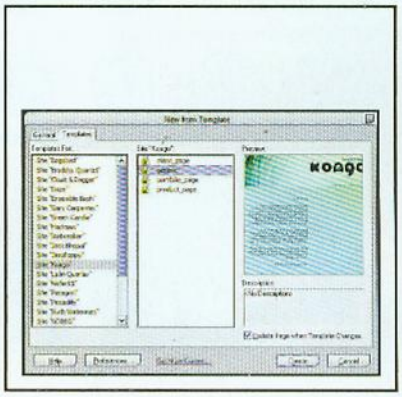
Add the finishing touch to the shopping cart with pages announcing the confirmation or cancellation of a transaction…

Order confirmation

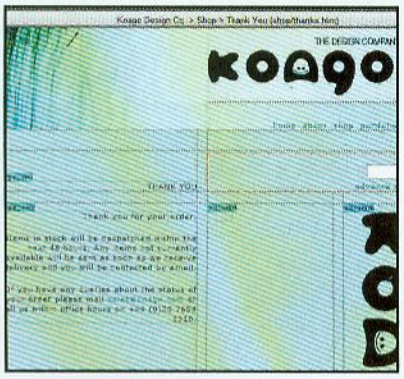
As well as notifying your customers of the status of their transaction through the diversion pages, you can also configure your PayPal account to automatically mail out receipts of a transaction. This provides the customer with your details in case they have any queries while they wait for delivery.



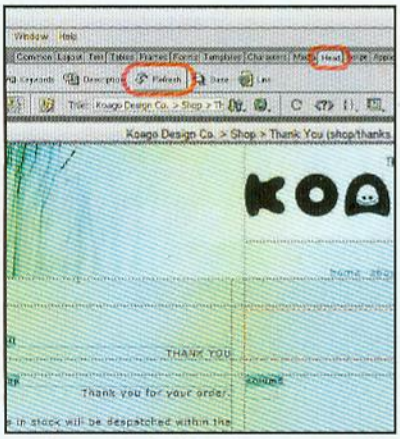
1 If you recall, as part of our Add to Cart button configuration we specified locations for pages that customers would be diverted to once a transaction had been completed. The final stage is to create the pages that have been referenced.



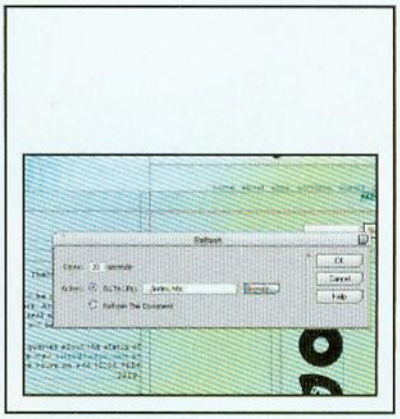
2 Create a new page using the generic template and give them the name thanks.htm. Save this file within the shop folder of the site to correspond to the location specified within the shopping cart buttons.



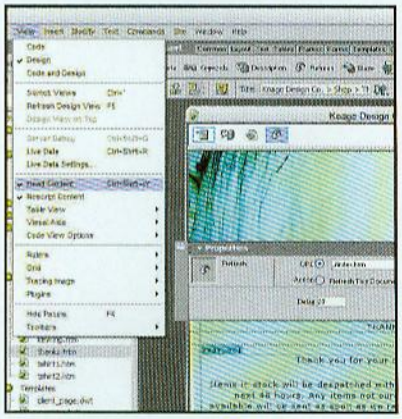
3 Replace the page and document titles and insert some text to thank the customer for their transaction. It's good practice to include contact details as a way of reassuring the customer of how they can get in touch if there are any problems with their orders.



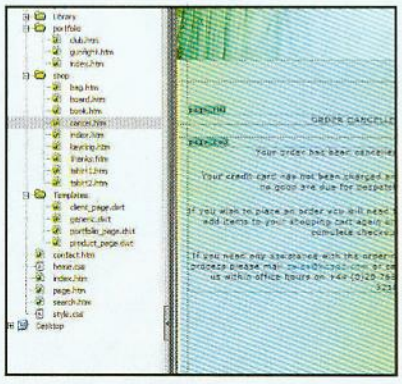
4 To get visitors back into the site, you can add a reload feature. This will refresh the page after a certain amount of time and load a chosen file, such as the homepage. Click on the Head tab within the main Insert bar and select the Refresh button.



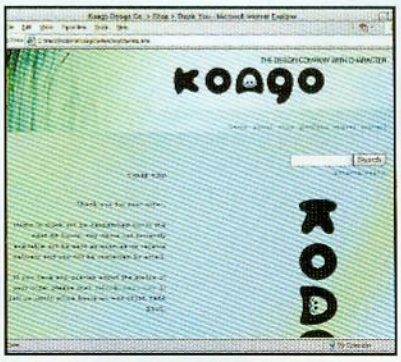
5 In the dialog box that appears, enter a delay time in seconds. This is used to determine how long the page will be held before taking the determined action, which we'll leave as Go To URL. Use the Browse button to locate your site's homepage and click OK.



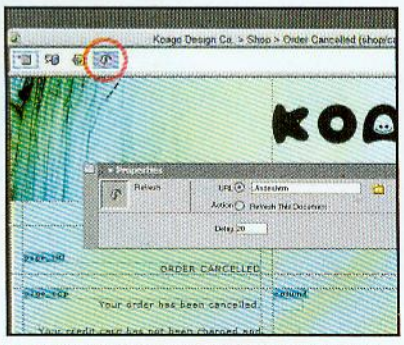
6 Such commands are stored in the invisible <head> of a document. To edit such a feature, choose View>Head Content to view icons at the top of your document that represent elements contained within it. Select the Refresh icon and the Properties Panel will allow you to make changes.



7 The same file can be used as the cancellation page. Simply amend the titles then copy and save as cancel.htm within the shop folder to match the location specified by the cart buttons.



8 Accessing these pages through your site will mean completing an order. You can check that the refresh function operates correctly by loading the file into your browser and waiting for the length of time you specified earlier.



9 Once the time has elapsed, the page should reload automatically to the requested page. If this fails, view your head content and check that the refresh configuration is set to Go To URL with a valid address entered, as opposed to Refresh This Document as shown here. ca>p

SECTION **SIX**

Publishing your site

With our site virtually complete, it's time to put the finishing touches in place so that koago.com can be accessed online...

Our site is now functionally complete. You may want to go through the pages and fine-tune a few graphics or reformat some text, but the basic site is now ready to be launched onto a live server for people to start using.

Before we blindly upload the site, though, there are a few tweaks needed under the bonnet that we haven't really considered up to this point. These are well worth carrying out to ensure our site is recognised by search engines, which will, most likely, be bringing a reasonable proportion of visitors to the site. After all, if the unique title of Koago doesn't appear at the top of a Google list then there's little chance of people finding our site or buying our products.

Meta tags are invisible components of a Web page that provide – at their most basic – a description and keywords relating to a page. Although every search engine has different criteria, it's good practice to put these in place to support visible content embedded into a page.

We'll also run through *Dreamweaver's* reporting tools to double-check our site has no obvious errors, such as broken links or missing images. Fortunately, our site is relatively small, and every page can be proofed prior to uploading. However, for larger projects this is an impractical task, and reports are often essential in finding such errors before your site visitors discover them for you.

All this needs to be done before our site can go live. We'll also make use of further *Dreamweaver* tools to ensure our local and remote sites are synchronised and up to date. Finally, once the site is live, we'll start to spread the word with some site promotion and search engine registration.

Files on disc
If you're jumping into the project at this point, the files needed can be found in the Tutorial\Section 6 folder on the cover CD.

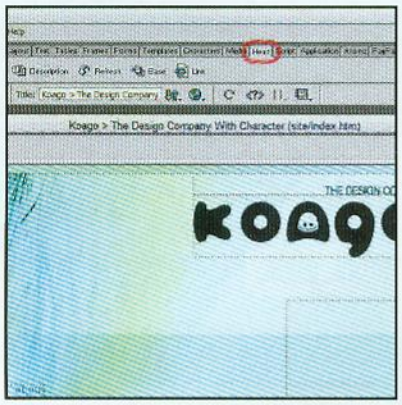
Illustration: Matthew Hems
Photography: Richard Dewdney

Part 1: Invisible identification

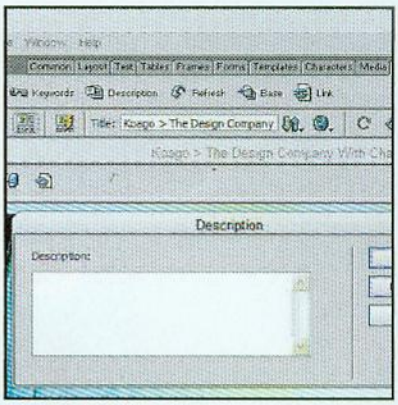
Add meta data to your pages to help out with search engines...

Make it relevant

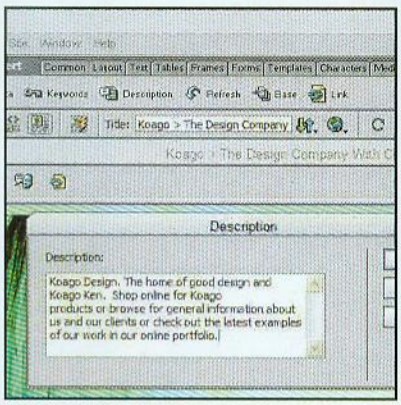
When adding meta data content, keep your information brief. Your keywords and descriptions should ideally be limited to 30–50 words, and should deal only with content relevant to your site. If meta information is too long, you might be penalised by those search engines that interpret extended entries as an attempt to raise your rankings by using generic phrases.



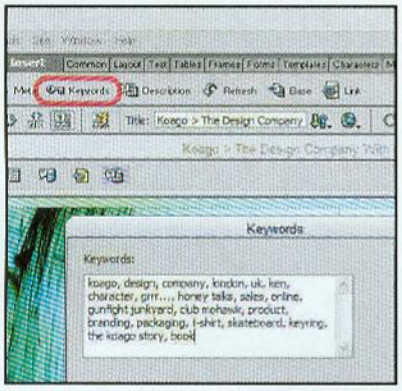
1 Open the homepage file – index.htm within the root of your site folder – and open the Head tab within the Insert panel to gain access to some regular meta insertion tools. These will help to make your site more recognisable to search engines.



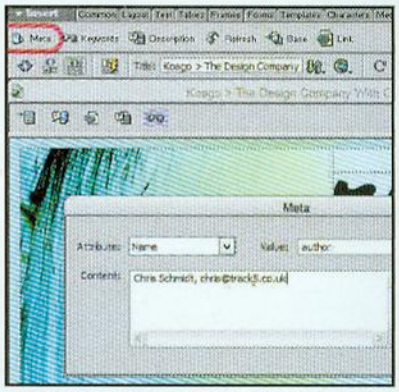
2 Check the View>Head Content option to open the visual icons for the elements we're about to add. Click the Description option from the Insert panel and an empty dialog box will appear for this feature.


Description
Description:
Koago Design. The home of good design and Koago Ken. Shop online for Koago products or browse for general information about us and our clients or check out the latest examples of our work in our online portfolio.

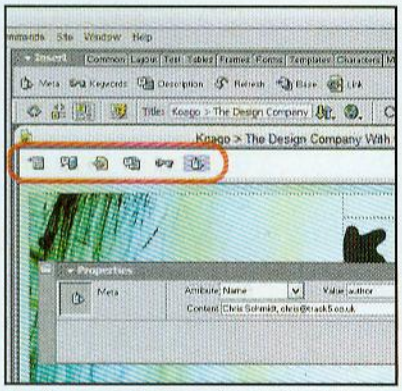
3 Enter a short description of the page's content. As this is for our site's homepage, we can include a general description of the site. When inserting the same feature on other pages, make sure you're more specific about that page's purpose.


Keywords
Keywords:
koago, design, company, london, uk, ken, character, grrr..., honey taka, sales, online, gunfight,junkyard, club mohawk, product, branding, packaging, t-shirt, skateboard, keyring, the koago story, book

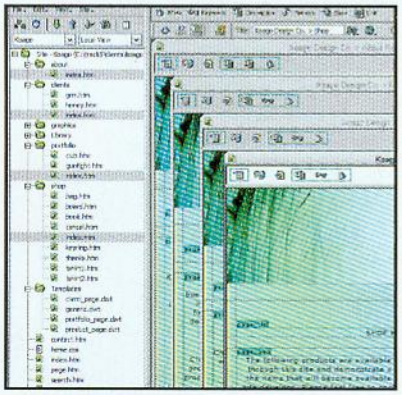
4 Click OK. Then click on the Keywords option in the Insert bar to bring up a similar dialog box. As the title suggests, this should include individual keywords relating to the page. The words should be separated by commas. Click OK when you've finished.



5 Although these are the default meta tags identified by search engines, you can place more specialised or specific tags using the blank Meta button from the Insert panel. Here, for example, we're embedding a tag that marks the author details.

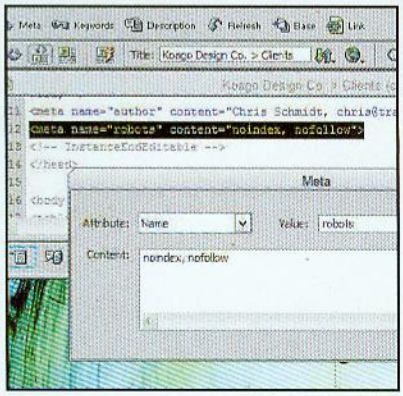


6 As you continue to create and add meta tags to your pages, you'll see icons appear within the page's Head area (if you've opted to make this visible). Clicking on these icons brings their attributes to view through the Properties Panel, where you can amend them.



7 In theory, meta tags should be embedded in all site pages, but this quickly becomes impractical even on our modest site. A more realistic option is to include them on major entry points, so run through the process for (at least) the About, Shop, Portfolio and Clients/Index pages.

8 In contrast to making pages available, you can also hide pages. Open the robots.txt file from the cover CD to get some idea of the syntax for such a page. Although such files are the subject of a whole other tutorial, you can get more information at **[w]** www.robotstxt.org on this.



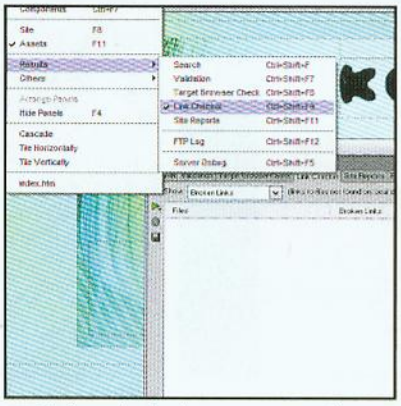
9 Alternatively, you could embed meta information into index pages of areas that should be hidden if required. For example, if we wanted to hide the contents of the clients pages, we would insert a tag as illustrated above into the opening page.

Part 2: Checking your site

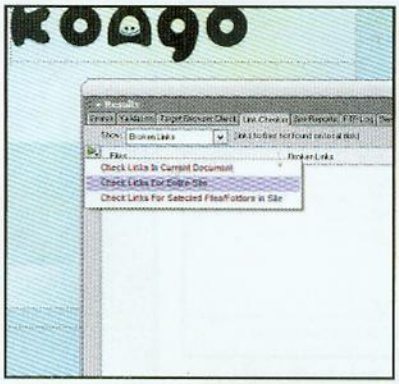
Use *Dreamweaver* reports to check for potential errors before the site goes live…

Check please

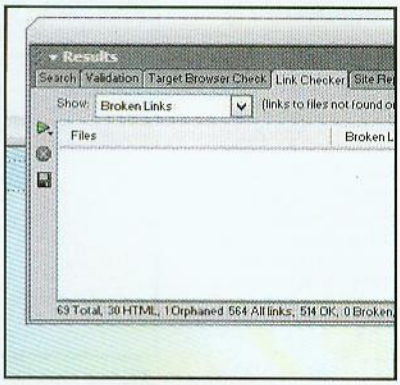
Testing your site is a process that should be carried out at every point of its development. The more testing you do as you build, the less your chances of coming across what could be potentially devastating errors. *Dreamweaver's* reports and results features can pick up some anomalous code-based errors, but if in doubt, check the pages yourself in the browser – this is the only sure-fire way of knowing whether your pages will be successful or not.



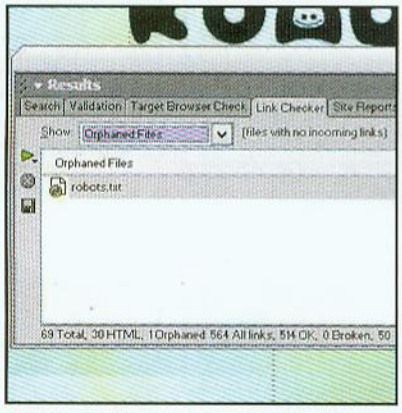
1 Before uploading the pages, you must be certain that any errors – such as bad links or missing images – have been weeded out. For this, you can turn to *Dreamweaver's* reporting tools. Choose Window>Results>Link Checker.



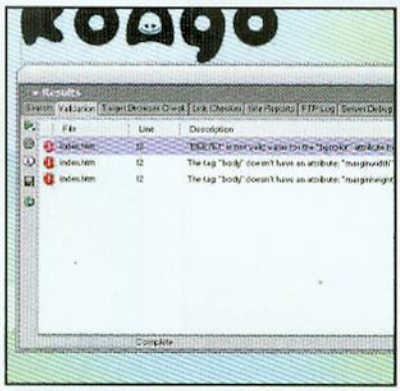
2 Select Broken Links in the Show drop-down menu and click on the green arrow in the left margin to select the Check Links For Entire Site option. *Dreamweaver* will search through your site's links checking for such errors, which it'll report upon completion.



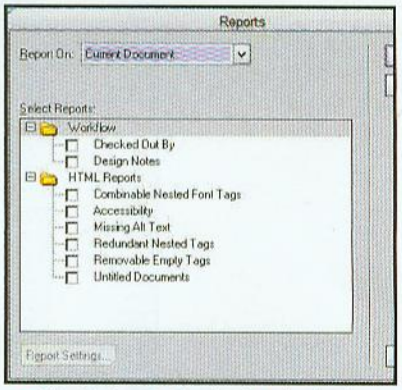
3 Your site should come back with no errors, but if any are found, follow the advice to rectify any rogue links. A summary of the results can be seen in the status area of the results – everything came out okay, although the one orphaned file mentioned here needs investigating.



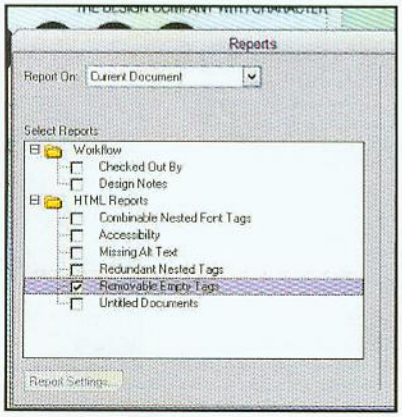
4 Use the Show drop-down menu to select Orphaned Files to display the file in question. Fortunately, the robots.txt file that we created along with our meta tags is supposed to be isolated, so we can ignore this reported problem.



5 The Validation and Target Browser Check tabs of the Results panel offer some insight to potential code problems. However, these should generally be used as guidelines because certain issues, such as the body tag errors above, may have been overruled by CSS external to the file.

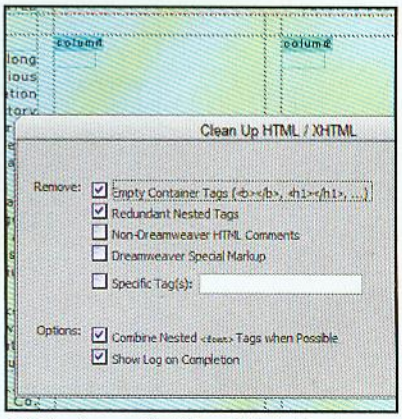


6 The Site Reports tab – also found under the Site menu – is a much more specific way of discovering common errors on individual files or throughout your entire site. Using this feature, you can determine particular issues that may cause your site problems when uploaded.



7 Running such reports throughout the development of your site is a good habit to get into. For example, open the main generic.dwt template file and choose to run Removable Empty Tags throughout the Current Document.

8 The report comes back with an empty anchor tag <a> or link contained within the editable region reserved for the secondary navigation. You might choose to delete this in order to avoid duplication in new files.



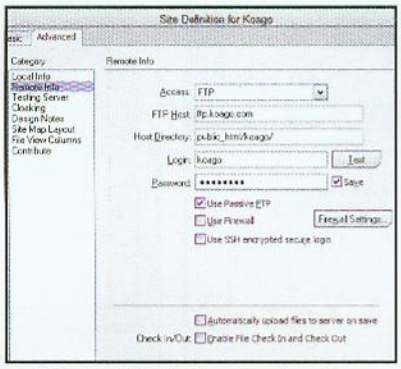
9 Another good habit to get into is cleaning up your pages' code via Commands>Clean Up HTML. This reduces the risk of errors associated with bad syntax, and also makes your pages much easier for the browser to render accurately.

Part 3: Publish and promote

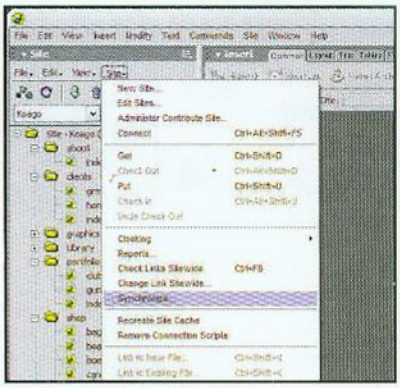
Upload your site and let the search engines know you've arrived...

Where am I?

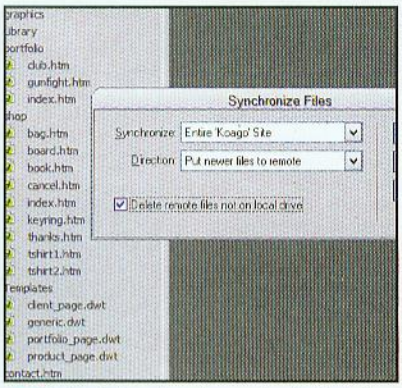
Just because you've submitted your site to every single search engine you could find doesn't mean you'll get listed automatically. Search engines can take minutes or months to locate and spider your site, but getting listed in results usually adds even more time to that. Be patient, and don't over submit your site or you could find you're inadvertently increasing the length of of time it'll take for your site to become available.



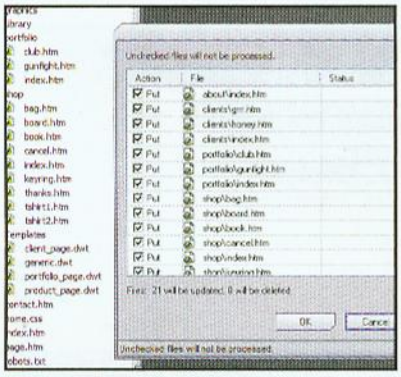
1 Prior to adding the Atomz Search engine, we configured the remote information for our site so that everything would be ready to load onto our server at this point. However, if you've jumped into the project since then, take a look at page 47 to see what's necessary.



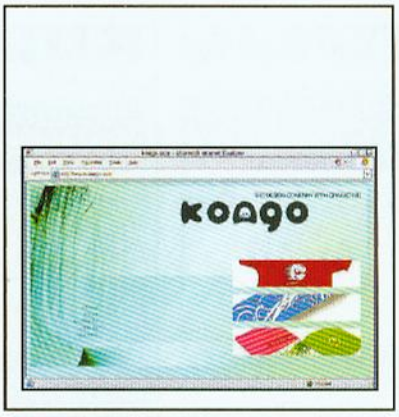
2 As our site's already partially uploaded, we can get *Dreamweaver* to synchronise our local version with that held on the server. Select all the files in the Site panel and choose the Synchronise option found on the Site panel's Site menu.


Synchronize Files
Synchronize: Entire 'Koago' Site
Direction: Put newer files to remote
Delete remote files not on local drive

3 In the dialog, choose to synchronise the Entire Koago Site, putting newer files to the remote server. You can also choose to tick the 'Delete remote files not on hard drive' option to remove any erroneous files that may be left behind.



4 Click the Preview button and *Dreamweaver* connects to your server to make the comparison. A report is returned, giving you options for files that need uploading and/or deleting. Take a look through and remove the tick from any that should be bypassed. Finish by clicking OK.



5 Congratulations. On completion of the transfer, your site is now live on the public server and accessible for anyone with your site's address. However, this may be a limited few so let's give things a helping hand...

6 The most obvious way of getting traffic is to achieve a high ranking in the major search engines. With a seemingly infinite number of sites to choose from, the best place to start is arguably Google, which powers many of the more established directory sites such as Yahoo!

7 Google's site submission page is buried away and best accessed through the site map. Once there, the process of letting the leading search engine know your site has arrived is to simply tell them your address and their spider will eventually crawl through and index your site.

8 Directory sites such as Yahoo! are much harder to get listed into because content is categorised by people, who take longer and are more selective about what's included. Adding your site to search sites like this often means browsing through to the most suitable listing page and submitting directly from there.

9 A basic ranking is just the start of a constant promotional process. Sites such as Microsoft's bCentral – **[w]** www.bcentral.com – provide various online tools to market, monitor and analyse your site's performance. With your site up and running, all you've got to do now is look after it... **ca>p**

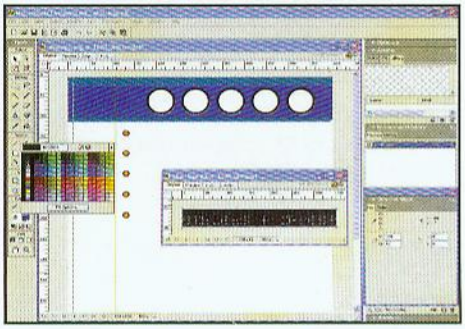
Faster Web Design

Increase the speed of producing your Website without compromising its quality. Simply follow these 30 essential Web design tips...

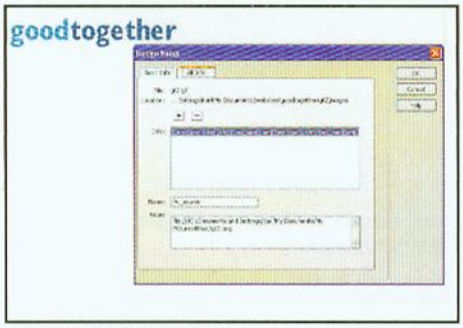
You're working to a tight deadline so you need to find the fastest way to get the best results. It's not all about taking shortcuts, though. Good practice is still an issue. It's no good finishing your site on time if it breaks on certain browsers, is full of broken links and has messy code.

Standards are still essential and planning is even more important. The aim is to streamline the process as much as possible without compromising on the results. Here, we've gathered 30 tips you can follow to speed up the production of your site, without sacrificing quality in the name of haste.

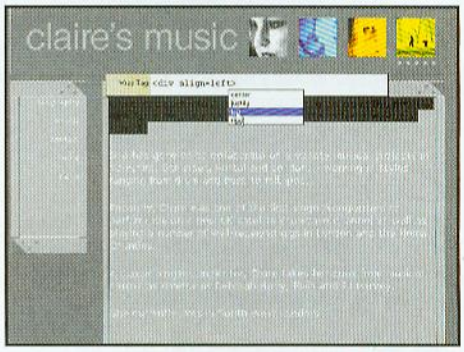
Expertise provided by Karl Hodge. You can contact Karl at [e] khodge@spodgod.com



Choose the Fill option and keep the mouse button held down to pick colour from anywhere on your desktop.



Location data about source files is stored as a design note in *Fireworks*' processed images. Edit them direct in *Dreamweaver*.



The Quick Tag editor in *Dreamweaver* enables you to edit code snippets quickly. With Auto-completion, you can start typing a tag and *Dreamweaver* will finish it.

FIREWORKS

Choosing colours

The Fill option in *Fireworks* enables you to choose colours from any other application that's visible on your desktop. This means you can sample colours directly from a page that you're working on in *Dreamweaver* or, if you have the Assets panel open, you can choose from the colour scheme section.

Symbols

Like *Flash*, *Fireworks* enables you to create Symbols, complete with optimisation and interactive features, that can be added multiple times to a project. A quick trick is to create a blank button using the button editor, then drag and drop multiple copies from the Library.

Integrated libraries

In *Fireworks*, create images or buttons as normal but export them with the extension .lbi to the Library folder of the site you're creating. When you open the site in *Dreamweaver*, your images will be there in the *Dreamweaver* Library ready to slot into the site.

Hidden messages

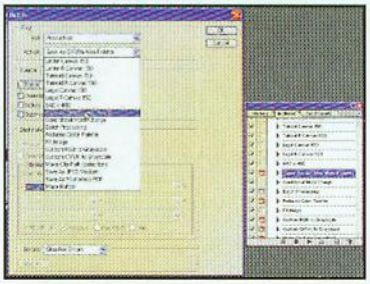
When you import an image from *Fireworks* into *Dreamweaver*, you also import 'meta data' about the file, including the location of the original source file. The meta data itself can be accessed and edited within *Dreamweaver* as a Design Note.

Cleaner code

When exporting slices, use *Fireworks* HTML options to simplify the code output as much as possible. Give slices unique names instead of allowing the program to name them for you, and try to export as a single table.

Batch processing

Batch-processing features in Macromedia *Fireworks* and Adobe *Photoshop* help to speed up the laborious work of converting images for the Web. Though you can individually optimise and save each image, it's much faster to create a script that will do the job for you. In *Fireworks* choose File > Batch Process. Add the files you want to process then click Next. Add 'Export' to the 'Include in Batch list', selecting the appropriate GIF or JPEG optimisation. You can also add a Rename command to put a suffix or prefix on each file as it's exported. When you've finished tweaking, click on Script and save your settings. The next time you need to convert a batch of images, simply double-click on the saved script instead. In *Photoshop*, go to Automate > Batch. Choose 'Production' as the Action Set — enabling you to choose to convert files to either the GIF89a or JPEG format. Set folders for source images and for the processed results and let *Photoshop* do the work for you.



You can record a batch process in *Photoshop* as an Action, further speeding up the process of converting images for the Web.

DREAMWEAVER

Start with a site

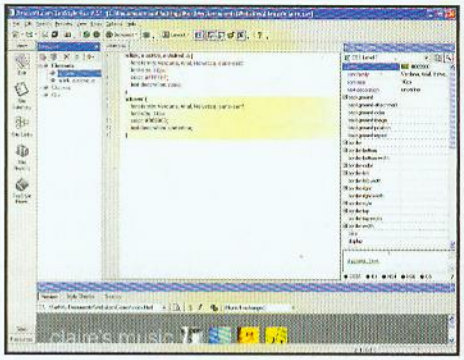
By default, *Dreamweaver* opens with the last site you were working on in the Site Window, plus a blank

Custom Dreamweaver objects

One of the main strengths of *Dreamweaver* is that it can be extended and customised easily by anyone with knowledge of Web design and HTML...

Using the Library allows you to insert components into pages without having to build them over again. If you find yourself constantly placing the same component into pages across different sites, why not turn it into a 'proper' *Dreamweaver* object? Start by going to the *Dreamweaver* folder on your computer, then the Configuration directory, and make a new folder called Custom inside the Objects sub-folder. Launch *Dreamweaver* and build the component on a new page, then save it in the new folder you just created. To make a unique icon for the object, make an 18x18 pixel GIF in an image-editing package like *Fireworks* and save it in the same folder. Make sure you give the icon the same name as the object file – for example, if your object is named logo.html then the icon should be called logo.gif. When you relaunch *Dreamweaver*, you'll have a new custom panel in the objects toolbar, with a new object ready to drag and drop into your pages.

The *Dreamweaver* Exchange is an archive of behaviours, scripts, extensions and custom *Dreamweaver* objects you can plug directly into the package. Access it through Commands>Get New Commands.



The <font> tag may seem to offer greater control over the formatting of content, but it can create cluttered code. For speed and ease, use standard paragraph formatting with CSS.

untitled document in the Document Window. If this is too much clutter for you, go to Edit > Preferences and click on General. Select 'Show only Site Window at Start-up' from the series of checkboxes.

Quick Tags

Quick Tags enable you to edit code without switching out of Design View. Select a point in your document that you need to edit directly and launch the editor with Ctrl-T in Windows or Command-T on the Mac. Press Enter/Return or click outside the QuickTag Editor to apply changes.

Templates

Always base your site around Templates, whether you build your own or use the ones bundled with *Dreamweaver*. A quick way to get started is to build your first page as normal, save that as a Template and then use that to create subsequent pages. Templates enable you to make changes site-wide by editing a single file.

Use the library

It's a good idea to get used to using the Library in *Dreamweaver*. As you build your site, keep the Library window open and add frequently-used objects like logos, navigation bars and even textual navigation strips. These items can then be re-used throughout your site.

Clear views

Click and drag on the tabbed section on a panel to tear it from its default group, then drag it to any other open panel group to dock it. This way, you can create your own custom panel arrangement containing the elements you use most frequently, while disregarding those you don't need for the current task in hand.

Hide panels

While working in *Dreamweaver*, you can quickly hide all panels by hitting F4, giving you a full view of the document you're working on.

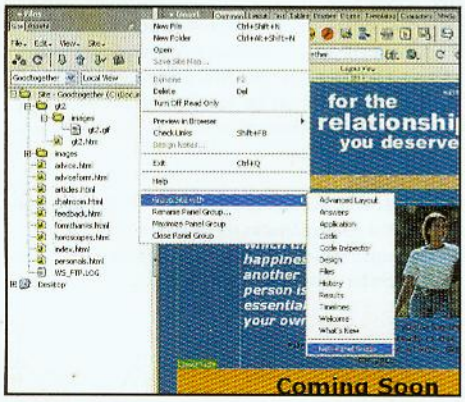
ADOBE GOLIVE

Low source

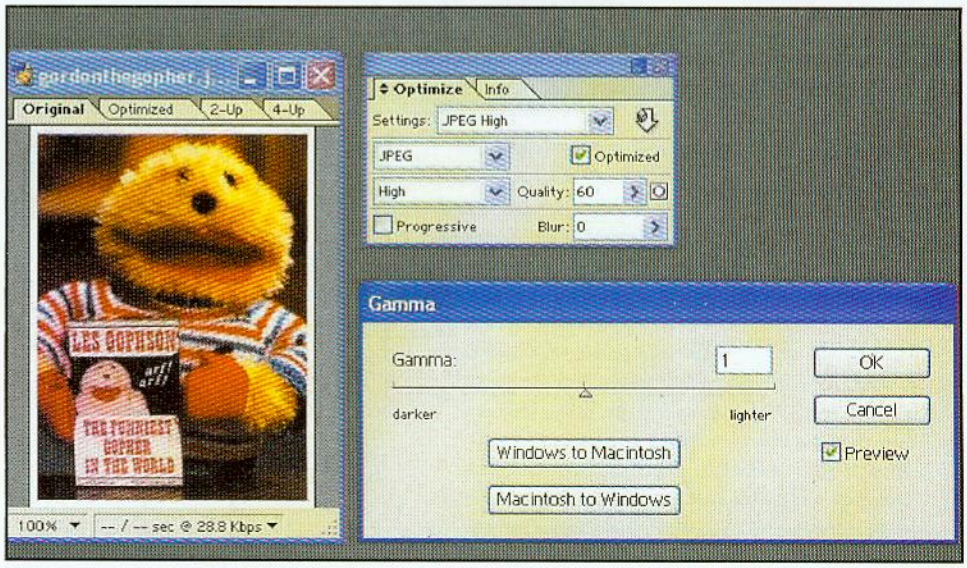
GoLive will automatically generate LOW SRC images for you from the default image on screen. Select your image then tick Low in the Inspector. Click on Generate to create the low-res image.

Cunning components

Components are *GoLive's* equivalent to Library elements – reusable chunks of code that you create like any other page. Building Components of a site first, then constructing the site by using them enables



Rearrange *Dreamweaver's* workspace to suit your needs. You can create and name new panel groups by going to the menu in any panel group and going to the 'Group x With' sub-menu.



PC gamma is optimised for the screen, Mac gamma is optimised for print. On Linux systems it's pitched somewhere in between... *ImageReady* enables you to adjust from Windows to Macintosh gamma and vice versa.

Speedy resources

Colour Schemer — What if you're fine with the coding part of page building but a bit of a dolt when it comes to the aesthetic side of things? You're not alone, and that's why there's a market for tools like Colour Schemer. The online version enables you to choose compatible colours for your Website in seconds. **[w]** www.colorschemer.com/online/

CSS Mill — A free online tool for generating CSS text styles — and that's it. A great utility when you're in a hurry and want to preview how a font will look on the page before you commit to code. **[w]** idlabs.org/css/mill/

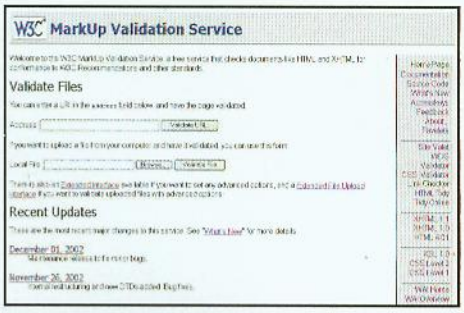
Blogger — Static sites are fast becoming a thing of the past. Web logging is one method of updating content dynamically, and the software is very flexible — it doesn't have to be used just for whiny Web journals. Blogger is the best known and most well-presented example and it comes with a bundle of ready-made templates. Your new site could be online within hours. **[w]** www.blogger.com

Free Layouts — Let someone else do the hard work, and download a free template from the thousands submitted. All you have to do is put in the content and you're ready to go. **[w]** www.freelayouts.com

Hot Scripts — Add dynamism to your site with free interactive scripts. With thousands of ASP, *ColdFusion*, PHP, CGI and JavaScripts ready to download, Hot Scripts is the Web's best site for plug-in dynamism. **[w]** www.hotscripts.com

See how fonts look with CSS Mill, a visual tool for generating CSS-styled HTML.

Free content management doesn't get much better than Blogger. Now owned by Google, you can expect it to become more feature-filled over the coming months.



You can check out how compliant (or uncompliant) your code is at [w] validator.w3.org. Don't worry if your page fails the test – very few pass the rigorous standard.

you to create pages that can be updated quickly and easily. Updating the original Component modifies all the pages that contain it.

No overlaps
Don't worry about overlapping areas in image maps with Adobe *GoLive*. The Image Map Inspector enables you to specify which hotspots take precedence.

Control content
After creating a new site in *GoLive*, make new folders in the site root folder for images, *Shockwave* files and other media. As you collect or create new media, place them directly in the correct folder. *GoLive* has its own ideas about where it wants to place content files — creating your own folders in advance gives you more control.

Custom elements
If you want to add a logo or other single element to multiple pages, you can select it within the page you're on and drag it to the Custom tab in the Palette. Items stored here are available to any site you work on.

WORKFLOW TIPS

Reuse images
Images can be used many times throughout a page with little impact on download and rendering time. Once an image has been downloaded, it's stored in the browser's cache on the user's hard drive and can be accessed more quickly.

Media resource
Create a Resource folder within the structure of your site and put in any files that may come in handy, such as text files with content, inspiring images, and *Photoshop* or *Fireworks* versions of the layout. You'll never have to hunt around for essential elements again.

Put it on paper
Even though you're trying to save time, work things out on paper first. Draw your layout, make lists of the types of media required and sort out the link relations between pages. An hour spent on these matters at the beginning of development saves time later on.

Easy updates
If your server supports Server Side Includes, persistent design elements within your site can be placed in your pages using SSI calls for easy updates. Place the HTML elements in a text file, then use the code
`<!--#include file="path_to_text_file" -->`
where you want to insert it. Give your page a .shtml extension and upload the text file along with your page.

Script snippets
Save frequently used snippets of script in a text file. When their time comes, open the file and cut and paste them directly into your document's code.

Site-wide styles
Use standard HTML paragraph names in combination with an external Cascading Style Sheet. That way you can change the formatting of your entire site just by editing one file. Forget the <font> tag.

COMPATIBILITY

Mac Gamma
If things look right to begin with, you'll save time in the development cycle. The normal Macintosh gamma setting of 1.8 creates images that may look dark on PC screens. Mac OS X users, go to System Preferences> Displays then select Colour. Click on Calibrate and go to stage 2, Set Up. Choose 2.2 Television Gamma — this is closer to the setting used by PCs.

Gamma emulation
If changing your default colour settings is too traumatic, you can use *Fireworks* to emulate the appearance of PC gamma while working on a Mac and vice versa. Go to View and choose the appropriate setting from the menu.

Web safety
Netscape Navigator uses a 216 colour 'non-dithering' palette for 8-bit images on both the Mac and PC. This is the closest thing there is to a universal 8-bit palette for the Web. GIF Images optimised with this 'Web safe' palette in *Fireworks*, *Image Ready* or any graphics package, should look fine at any resolution on any platform.

Browser standards
HTML isn't as standard as you might think. *Navigator* and *Internet Explorer* both contain proprietary tags that the other doesn't understand. In *Dreamweaver* you can check target browsers to make sure your pages will display correctly using File >Check Target Browsers.

Find a friend
Nothing beats checking things out yourself. Look at your pages in the latest versions of *Netscape Navigator* and *Internet Explorer* at least. Check them on both Mac and PC, even if you have to nip over to use a friend's machine. *Netscape 7.0* may be more compliant with W3C standards than *Internet Explorer* — but more people use the latter. cap

The brief for this magazine article was simple: show how open source software can help you develop a shopping cart, how full books have been dedicated to building shopping carts and how they still don't cover the topic thoroughly enough. So what can we hope to cover in 3000 words? Well, quite a lot actually. This article has been broken into two separate sections. In the first section, we look at building a simple shopping cart using PHP. Although this cart will only work for the most basic of stores, it'll help you understand the basics of how shopping carts work. In the second section, we look at *os Commerce*, a well-written PHP shopping cart, which, best of all, is free to download.

Part One: *PHP makes building a shopping cart easier*

PHP – or PHP Hypertext Preprocessor to give it its full title – is quickly becoming the Web-scripting language of choice. It's particularly suited to designers making the shift over to Web development. Even if you don't come from a programming background, you'll pick up the basics of the language quickly.

Welcome to 'Geekposters'. Imagine you're a poor geek, so in order to keep your coffee and Cheetos habit going, you've decided to open an online store selling a fine collection of posters all with different geek-like sayings. So what components do we need to build our cart? We must track users as they add items to their basket, and we need to store this information. Tracking users is simple enough – we can use either cookies or session cookies. In this example, we'll use sessions. To understand why, we need to look at the difference between the two.

· Cookies are small text files that are stored on the user's machine, usually in the format of name=value. These cookies can be given a lifetime (the length of the time the cookie will last). This is ideal if you want to store information between visits to a site. A user can visit a site days, even months, later and any information stored in the cookie will be available (as long as the cookie hasn't expired). A typical use for a cookie is when you sign into a forum or message board and tick the box that says 'remember me'.

· Sessions, on the other hand, are text files that are stored on the server. Each file is given a unique id (session ID), which associates it with a particular user. A session will only last as long as the user's browser is open. As soon as the browser's closed, the session information will be destroyed. Simply put, a session will not store information between visits.

Now, while at first it may appear that cookies would be better for the job, this isn't the case. First of all, not all users have their browsers set to accept cookies, and, after all, we want our online shop to be accessible to everyone. Secondly, as the file is stored on the user's machine, it means it's easier to tamper with the information stored within it. Therefore, sessions offer us better security too.

The next decision we need to make is how to store the user's information. One of the great things about PHP is that it offers native connectivity to many Regional Database Management Systems (RDMS). Chances are, if your host offers PHP

Illustration: Stephen Johnston [w] www.phantomresearchfoundation.com

Working with Dreamweaver MX

IntelliCART MX is used by Macromedia Sales Engineers worldwide to demonstrate what is achievable with *Dreamweaver* PHP...

An IMX presentation formed part of David Demmings' (DMX product manager at MM) keynote speech at the PHP Conference in Frankfurt last year.

More than 2000 Web design companies are currently using IMX worldwide. It's being used by Television Shopping Channel Thane Direct in the UK, Germany and Canada ([w] www.thanedirect.co.uk, www.thane.de and www.thane.ca).

A single IMX licence costing around £95 ($148.75) enables you to create an unlimited number of e-commerce Websites without further charge.

IMX adds 37 new Server Behaviours and six new Server Objects to *Dreamweaver MX*, and is used in conjunction with the existing *Dreamweaver MX* tools.

Additional plug-ins are available that add support for WorldPay, Authorize.net and 2Checkout.com payment processors, and are available at no additional cost to customers. IMX can also be integrated with other payment processors easily.

A dedicated customer support portal provides IMX Support, providing quick and easy access to support forums.

IMX is designed to be integrated with any Website, and virtually any database structure, giving you complete control over all design aspects of your site.

as an option, it'll offer MySQL as well. So let's take advantage of this and use MySQL. Although many new users to PHP think using a database will be harder than, say, using a flat text file system, this isn't the case.

Now that we've decided that we'll use a database, we need to decide on its structure. Be warned that under normal circumstances a lot of planning needs to go into the design of your database. Data normalisation is an important part of any project. However, in this example we'll keep the database structure as simple as possible. So keeping this in mind, we'll only have three tables.

my_items This will store information on the goods for sale, description and price along with a unique id so that we can identify each product.
my_orders This will contain the orders that have been placed, but have not been processed yet.
my_completed This will contain the completed order along with the user's shipping address and contact details.

Finally, we need to decide on the general layout. We need a page that displays all the items we have for sale. We also need a page that'll contain the PHP script that will process the information. We then need a checkout page that will list the items and allow the user to change quantities. Finally, we need to process the user's details along with their credit card information. So, in total, we require four pages, which we'll call:

cart.php This page displays the items we have for sale and enables the user to select the quantity.
add_cart.php This page contains the PHP code to parse the information from cart.php and add it to the database.
checkout.php This page displays the contents of the user's basket and allows them to either alter the quantities they have ordered or enter their information to process the order.
process_order.php This page gathers the user's order and personal information and stores the completed order in the database.

Let's look at the PHP code on each page and walk through what it does. All code examples are well commented, so we can follow it from those. First of all, we'll look at cart.php because this is the page that displays the items we have for sale.

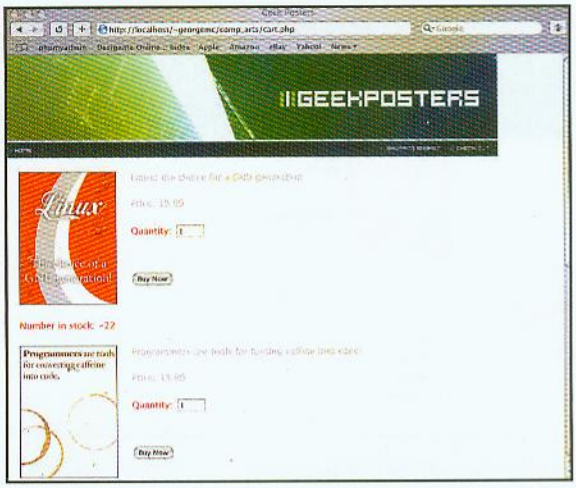
The code on this page is broken into two blocks. The first block on the page is similar to the other pages. In it, we open the connection to the MySQL server and select the database. The only difference with this example from other pages is that we use a condition block to check and see if our session is set; and if not, we create it and assign it a value.

```
<?php
//Start our session
session_start();
//Check to see if the session has been registered
//if it hasn't register it and assign the user an ID
if(!isset($_SESSION[user_id])){
$_SESSION[user_id] = time();
}
//Open up the connection to our database. Our username is comp_arts
// and password is password * change these to suit your own values
$connection = mysql_connect("localhost", "comp_arts", "password") or
die("Error: Failed to connect to Database");

//select the database
@mysql_select_db("my_cart", $connection) or die("Error: Cannot select
database");
?>
```

The second block pulls the information from the database and displays it on the page.

```
<?php
//Query the database and return information
$result = @mysql_query("SELECT * FROM my_items");

//print out the html code to start the table
print("<table width=\"400\" border=\"0\" cellspacing=\"12\">");

//loop through our result and display the
// items on the page
while($row = mysql_fetch_array( $result )) {

print("<form method=post action=add_cart.php>");
print("<tr><td rowspan=4><img src=\"" . $row[image] . "\"></td>");
print("<td>" . $row[description] . "</td></tr>");
print("<tr><td>" . $row[price] . "</td></tr>");
print("<tr><td>Quantity: <input type=text name=quantity
size=5></td></tr>");
print("<tr><td><input type=submit name=submit value=\"Buy
Now\"></td></tr>");
print("<tr><td colspan=2>Number in stock: " . $row[stock] . "</td></tr>");
print("<input type=hidden name=postID value=" . $row[id] .">");
print("</form>");

}
//print out our closing table tag
print("</table>");
//close the connection to the database
mysql_close($connection);
?>
```

Next, we'll look at add_cart.php. This is the script that processes the information from cart.php and adds it to the database. We re-use



the same code from cart.php to open the connection to the MySQL server and select the database.

Our next block of code provides some basic error checking while processing the information; if there are no errors, then it places the information into the database and displays a 'thank you' message. If there is an error while processing the order, it displays an error message.

```
<?php
 //Query the database and return information
 $query = mysql_query("SELECT prodid, quantity FROM my_orders WHERE
prodid = $_POST[postID] and user_id = $_SESSION[user_id]") or
die(mysql_error());

 //If the user has already ordered the item, only update the quantity
 if(mysql_num_rows($query) != 0) {

 mysql_query("UPDATE my_orders SET quantity = quantity +
$_POST[quantity] WHERE prodid = $_POST[postID] and user_id =
$_SESSION[user_id]") or die(mysql_error());
 mysql_query("UPDATE my_items SET stock = stock - $_POST[quantity]
WHERE id = $_POST[postID]") or die(mysql_error());

 } else { //order doesn't exist so insert it into the database

 $result = mysql_query("SELECT * FROM my_items WHERE id =
$_POST[postID]");

 while($obj = mysql_fetch_object($result)) {
  mysql_query("UPDATE my_items SET stock = stock - $_POST[quantity]
WHERE id = $_POST[postID]") or die(mysql_error());

  mysql_query("INSERT INTO my_orders(user_id, prodid, description, price,
quantity, image, time ) VALUES($_SESSION[user_id], $obj->id, '$obj-
>description', '$obj->price', '$_POST[quantity]', '$obj->image', now())") or
die(mysql_error());
   }
}
//Perform a check to see if information has been entered correctly
//if it hasn't, display an error otherwise display a thank you message
 if( mysql_affected_rows( ) != 1 ) {
  print("Error: Failed to add item to basket. Please try later"); //error message
to display on failure

 } else {

  print("Thank you, the item has been added to your basket<br />");
  print("<a href=\"cart.php\"><img src=\"images/continue_button.gif\"
width=\"200\" height=\"50\" border=\"0\"></a><br /><br />");
//message to return on success
  print("<a href=\"checkout.php\"><img
src=\"images/checkout_button.gif\" width=\"200\" height=\"50\"
border=\"0\"></a>"); //message to return on success
  }

?>
```

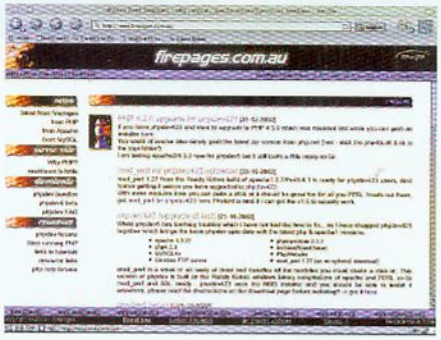
Now that the user has placed their order in the database, we need a way for them to view it and make any alterations. This is the job of checkout.php. Again, we re-use the same code to open the connection and select the database. Our next block of code pulls the order from the database and displays the item, quantity and cost. We also give the user the opportunity to update any quantities. Finally, the total cost of the order is displayed. If the order is correct then the user can enter their personal details into a form and place the order.

```
<?php
 $result = mysql_query("SELECT * FROM my_orders WHERE user_id =
$_SESSION[user_id]");

 $total = 0;

 print("<table width=500 border=0 cellpacing=5 cellpadding=0>");

 while( $row = mysql_fetch_array($result) ) {

  print("<form method=post action=" . $_SERVER[PHP_SELF] .">");
  print("<tr><td rowspan=4 align=center><img src=\"" . $row[image] .
"\"></td>");
  print("<td>" . $row[description] . "</td></tr>");
  print("<tr><td>" . $row[price] . "</td></tr>");
  print("<tr><td>Quantity: <input type=text name=quantity size=5 value=\""
.$row[quantity] . "\"></td></tr>");
  print("<tr><td><input type=submit name=submit value=\"Update
Quantity\"></td></tr>");
  print("<tr><td colspan=2 align=right>Total: " . $total_price = ( $row[price] *
$row[quantity] ) . "</td></tr>");
  print("<input type=hidden name=postID value=" . $row[prodid] .">");
  print("</form>");

 $total = ( $total + $total_price );
 }
//register a session that will store the total value
$_SESSION[total] = $total;
```

Starting out with PHP

Learning a new language isn't easy. Make the most of these resources...

Hopefully this article has sparked an interest in PHP. If you're a designer hoping to add a few extra skills to your CV, you'll find PHP a welcome addition. However, as with all new languages, getting started is the hardest step. There's a wealth of books available on this subject, ranging from the complete beginner to advanced topics.



One of the best places to find information is on the official PHP Website at [w] www.php.net. There are also a number of forums that will help you if you have any questions. Sites such as [w] www.designateonline.com/forum have sections dedicated to PHP.

Another common problem most newcomers face is when they try to set up Apache/MySQL and PHP on their own machine for testing their scripts. The good news is there are a few programs available that will install all these for you. Probably the best available is phpDEV, which can be downloaded from [w] www.firepages.com.au . If you use a Mac then you can download an installer program from [w] www.entropy.ch/software/macosx/welcome.html

```
print("<tr><td colspan=2> </td></tr>");
print("<tr><td colspan=2 align=right>Total Cost: ". $total. "</td></tr>");
print("</table");

//close the connection to the database
mysql_close($connection);
?>
```

Finally, we have the page that processes the order. Again, we use the same code to connect to the database. Now that we've made the connection, we enter the user's information. Finally, we use the mysql_affected_rows() function to check to see if the information was entered. If it wasn't, then we display an error message; otherwise, we display a thank you message.

```
<?php

//Insert information into the database
 $result = @mysql_query("INSERT INTO my_completed(user_id, name,
address, city, postcode, telephone, email)
 VALUES ('$_SESSION[user_id]', '$_POST[name]', '$_POST[address]',
'$_POST[city]', '$_POST[postcode]', '$_POST[email]',
```

```
'$_POST[telephone]')") or die(mysql_error());
//print out some line breaks
print("<br /><br />");

//check the amount of rows affected by the query
//if it equals one then the info has been entered
if(mysql_affected_rows() == 1 ){

 print("Thank you for your order your order id is " . mysql_insert_id());

 } else {

 //The rows affected does not equal one
 // this means there has been an error
 print("Error:Failed to enter your order in the database. Please try again");

 }

//close the connection to the database
mysql_close($connection);

?>
```

In the above case, we've used the user_id as a key, so we can reference the order in the my_orders table.

Part Two: Why re-invent the wheel?

So far, we've looked at building a simple shopping cart. This was ideal when we only had a few items for sale, and we didn't require any credit card processing. Now we have another option. Our Geekposters site is a hit and we've decided that we want to sell all sorts of other geek paraphernalia through it. It's obvious that our existing shopping cart will not be big enough for the job. So we now have two options.

Option one: we could build our own large-scale solution. The problem with this option is that development time would be lengthy. Chances are, once complete, the application would require constant maintenance, updates, bug fixes, and security problems, all of which would consume more time. However, the good news is that there's a second option. There are a lot of PHP developers who have written some great applications, many of which are offered free under one of the various free software licences available. One of these is *os Commerce*. This application has all the features that we need for our new and improved online store. So we'll download it and install it so we can start selling.

The latest version if available from [w] www.oscommerce.com. In this example, we're using version 2.1.

Once you've downloaded the application, you need to unzip the files. This will leave you with a folder called tep-pr2.1. Inside this folder, there are another two folders: one called tep_catalog-pr2.1 and the other called tep_admin-pr2.1. Enter the tep_catalog folder – within this you should see a folder called catalog.

First, you need to open up the applications_top.php file. This is located in the includes folder, which can be found in the catalog folder. If your host is running 4.2.0 or above, or has register_globals set to 'off', then you'll need to add a line of code to this file. Look for the following comment at the top of the file:

```
// for internal use until final v1.0 version is ready
```

Directly above this, add this line:

```
$PHP_SELF = $_SERVER[PHP_SELF];
```

With this done, you can make the rest of the changes. On line 15 and 16, you'll find the following code:

```
define('HTTP_SERVER', 'http://localhost/');
define('HTTPS_SERVER', 'https://localhost');
```

You need to change it from http://localhost to your domain. If you're not using SSL then you can leave the bottom line untouched. However, if this is the case, you'll need to change the line below from:

```
define('ENABLE_SSL', 1);
```

to:

```
define('ENABLE_SSL', 0);
```

Then you need to set the currency. Go to line 32 and change this bit of code:

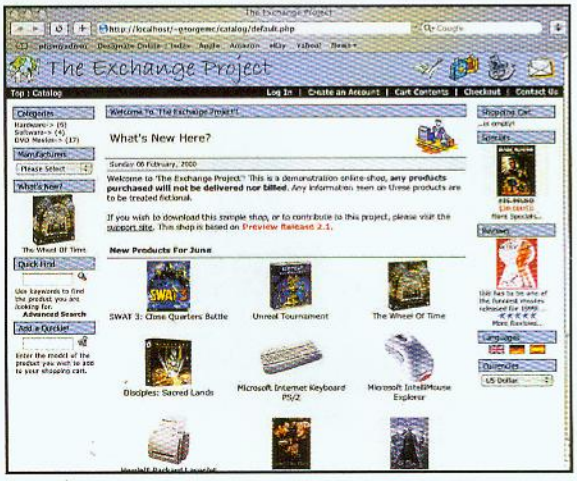
```
define('DEFAULT_CURRENCY', 'USD'); // use the code
```

to:

```
define('DEFAULT_CURRENCY', 'GBP'); // use the code
```

Finally, go to line 88 and you'll see this block of code:

```
// define our database connection
 define('DB_SERVER', '');
 define('DB_SERVER_USERNAME', '');
 define('DB_SERVER_PASSWORD', '');
 define('DB_DATABASE', '');
 define('USE_PCONNECT', 1);
```



If we use the same username and password as our simple shopping cart, we would change it to:

```
// define our database connection
 define('DB_SERVER', 'localhost');
 define('DB_SERVER_USERNAME', comp_artst');
 define('DB_SERVER_PASSWORD', password');
 define('DB_DATABASE', 'shop');
 define('USE_PCONNECT', 1);
```

We now need to upload the entire catalog folder to our Web directory. With this done, we need to create the database and import the SQL file, which is contained within the tep_catalog-pr2.1 folder. If you or your host has phpMyAdmin installed then you can use this for the task. If you don't have access to phpMyAdmin then you'll need to upload the SQL file into the home directory and use the command line to create the database and import the file. In this example, we've created a database called 'shop'.

You can now close the application_top.php file and point your browser at [w] yourdomain.co.uk/catalog/default.php. You should now see the default install page for *os Commerce*.

Nearly done. The last thing you have to do is copy the admin folder, which is located in the tep_admin-pr2.1, to the Web root directory. Once you've done this, open up the folder included within the admin folder and then open the file application_top.php. We need to make the same changes as we did to the application_top.php file that is located in the catalog/includes folder. The only difference is we don't need to specify an HTTPS server.

The only thing left to do is to set the ownership for the images folder within the catalog folder so we can upload images to it. You can use your FTP client or do it from the command line. If you're using the terminal, you need to enter the following commands:

```
cd catalog  #move into the catalog folder
chown -R nobody.nobody images #change the ownership of the images folder
```

Finally, and most importantly, we need to password protect the admin directory. The best way to do this is by using an htaccess file.

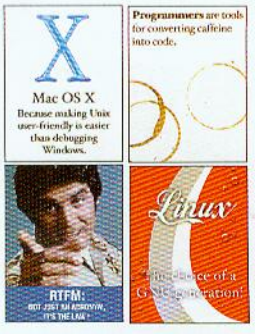
We can now start to set up shop by visiting:
[w] yourdomain.co.uk/admin/default.php

Deploy your cart

You should now have the knowledge to start selling online...

Hopefully this article has encouraged you to start building your own cart, or to download and install *os Commerce*. Either way, we hope it helps you to start selling online. During the first part of the article we looked at building a simple cart.

Because of constraints over the overall size of the article, we covered the more important parts of building the cart. Because of this, some simpler parts of the cart were left out. However, all the code for this article is available for download from [w] www.experimentalmonkey.co.uk/ca, including the files we detail, the admin files for our simple shopping cart, along with all the images used and an SQL file that will help you create the database. Also included in the download are install instructions.


Mac OS X
Because making Unix user-friendly is easier than debugging Windows.
Programmers are tools for converting caffeine into code.
RTFM:
Linux

Finally, we would like to thank Mike (mail@shunk.co.uk) for designing the logo and header for the sample pages, along with the posters. Given that he only had a few hours' notice to produce something, he's done a great job. We may even sell the posters for real.

COMPUTER arts projects INTERACTIVE

On the CD

Your guide to the free extras given away on this issue's cover disc...

PRE-BUILT WEB PAGES

Free Dreamweaver templates

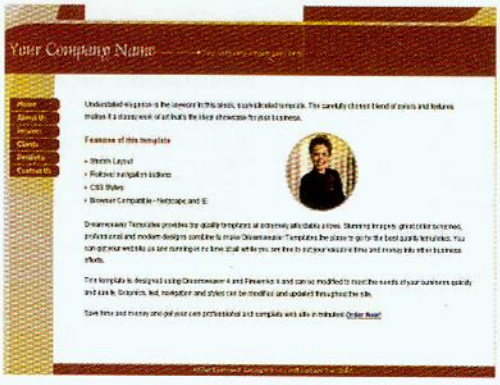
Get your site off to a flying start with five *Dreamweaver* templates – worth $150 – courtesy of dreamweaver-templates.net...

When you're building an online store site, your primary concern is the e-commerce functionality that enables you to start making money. Pre-built *Dreamweaver* templates from www.dreamweaver-templates.net give you a much needed head start. Choose from a series of design styles to create a fully featured Website without having to get your hands dirty with HTML source code.

Each template contains a core DWT (*Dreamweaver* template) file along with portable network graphic (PNG) format image files, cascading style sheets and JavaScript code files. The templates are easy to customise using *Dreamweaver* and *Fireworks MX*, and there are plenty of designs to suit your particular needs.

On this month's cover disc you'll find five commercial Web page templates worth $150, which can be used in your own Websites. Readers of *Computer Arts Projects* are also eligible for a 25 per cent discount by visiting [w] www.dreamweaver-templates.net/futurepublishingoffer.htm

www.dreamweaver-templates.net



FULL PC PROGRAM

EROL 3 Small Store

An online store could be just the boost your business needs. *EROL 3* is a fear-and-fuss-free way to get your products seen

WORTH £116

An online store is one of the most effective ways to get your business noticed. *EROL 3* enables you to boost your business potential and create this additional sales forum yourself. Customers can then tap into your online store, search through your product list and place orders.

The whole process has the potential to be fraught with difficulties, but the *EROL* software dispels any fears you may have. It's easy to keep track of orders, constantly monitor stock levels and even choose your own payment options. Another useful aspect is that you can publish your e-store on CD ready to be sent to customers who are not online.

With *EROL*, you can create a template-based e-store with a maximum of ten products. This makes it ideal for the smaller business, while larger businesses will find it useful as a sample version before going further.

EROL's templates can be used as they are or edited from within the program itself. The software also integrates with Macromedia *Dreamweaver* for advanced editing.

Many people are understandably worried about the potential problems involved in accepting online payments, but *EROL* makes the whole process very simple. There are a number of payment options, including receiving payments directly or going through a third party. This program can also bring peace of mind to both you and your customers by offering secure, encrypted emailing options.

There are a number of useful electronic manuals available for download from the *EROL 3* Website, covering the Build Manager, keeping on top of orders with the Order Manager and using the Customer Manager to keep a track of customer details. You can download these help files from [w] www.erolonline.co.uk

Upgrade to another mighty EROL edition

For those businesses with serious online sales in mind, check out *EROL*'s upgrades...

EROL 3 Business is suited to the slightly larger company (an upgrade from *Small Store* costs £329).

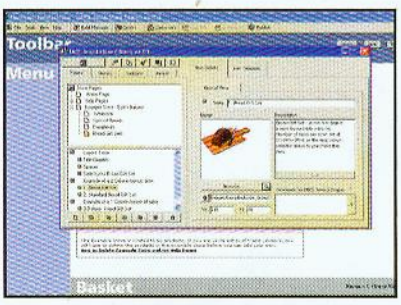
EROL 3 Developer is capable of managing multiple EROL stores from a single installation (an upgrade from *Small Store* costs £899).

EROL 3 Vendor is for people who only need to manage Order and Customer information (£169).

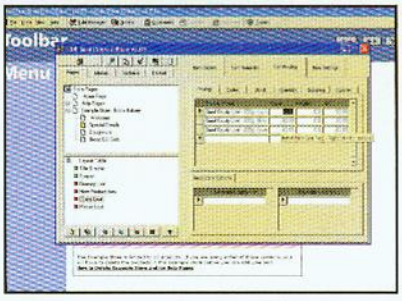
To find out more about any of these products, visit [w] www.erolonline.co.uk or call 0870 749 9141.

www.erolonline.co.uk

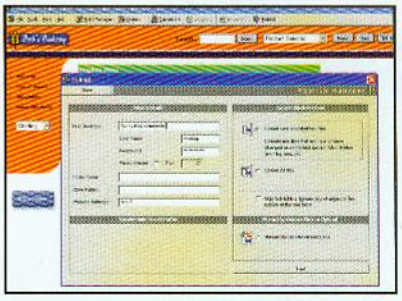
Getting started with EROL 3 Small Store



1 *EROL Small Store* comes complete with an example store: Bob's Bakery. By having a look at this store, you can see how the program works and adapt it to suit your own requirements. Launch the program, click on the Build Manager, then click on the Example Store. You can then see how it's put together and make any changes you want.



2 You can enter or change details for up to ten products by clicking on any of the red items in the bottom-left of the screen. Use the tabbed interface to set options such as cost, stock levels, shipping costs and more. You can then use the Menus, Toolbars and Basket tabs to change the look of your online store.



3 You can check how your site will look by clicking on the Preview Changes button in the Build Manager. Check through the site and ensure that it works as expected. If you're happy with the results, you can click on the Publish button. Enter your hosting information and click on Next to make your shop go live.

BOOK REVIEWS

FASTER SMARTER WEB PAGE CREATION

Author **Mary Millhollon & Jeff Castrina** Publisher **Microsoft Press** Price **£15.99** ISBN **0-7356-1860-7**

Aimed at beginners, this book offers an overview of Web page design that's as comprehensive as it is accessible. Solid foundations are laid in early chapters, explaining how the Internet and Web pages work together. If you're a complete novice, you'll learn all you need to know before you get your hands dirty with HTML coding.

Sample Web pages are constructed with Notepad, *Word* and *FrontPage*. This clearly illustrates that hundreds of pounds don't need to be spent on sophisticated Web design software. The step-by-step approach enables you to build your knowledge of each HTML tag sequentially until you have a complete Web page, and these steps later serve as a good reference work.

The strength of this book is the practical way in which it teaches you how to hand-code HTML pages with nothing more than a simple text editor. Using design packages shields the user from the underlying code and consequently undermines a deeper understanding of how their Website is constructed; this book offers a hands-on approach that'll serve you well as your skills improve.

VERDICT **4**/5

DEEP SITES

Author **Max Bruinsma** Publisher **Thames & Hudson**
Price **£18.95** ISBN **0-500-28384-2**

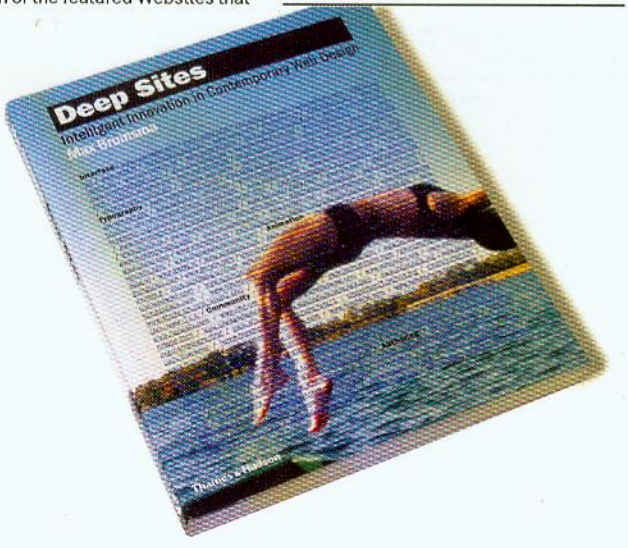
The Web has grown into a diverse infotainment space that contains information presented in a plethora of different formats. This book provides some guidance on how today's Websites shape up.

The first chapter is on interface design; subsequent sections cover typography, animation, community and authoring. Each chapter opens with an essay that looks at these topics before showcasing some of the best the Web has to offer in each category.

The five subjects are looked at in great depth, but it's the detailed dissection of the featured Websites that makes this book stand out from the crowd. A critique of each site is offered, enabling you to understand the design choices that were made during the site's construction.

This book is a bold attempt to show the reader how the Internet is being used today to showcase cutting edge sites. Anyone who's in the process of building a new Website will relish this book, because it offers a snapshot of current design thinking as well as page after page of inspired – and inspiring – results.

VERDICT **5**/5



ABSOLUTE BEGINNER'S GUIDE TO CREATING WEB PAGES (2ND EDITION)

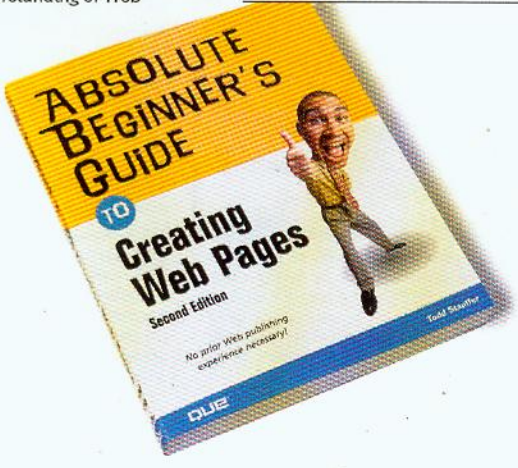
Author **Todd Stauffer** Publisher **Que**
Price **£13.99** ISBN **0-7897-2895-8**

If you want to learn how to build Web pages by getting your hands dirty from day one, then this is the book for you. It offers a solid overview of all the technologies currently being used in Web page creation. It doesn't, however, purport to offer any guidance on specific graphic design tools, such as *Dreamweaver*, instead focusing on the code that underlies these tools.

You won't be able to build a rudimentary Website in a day with this guide, but there are other books specifically geared towards that goal (not to mention this issue of *Computer Arts Projects*). What you'll gain is a thorough understanding of Web technologies and how you can use these to produce engaging and useful sites.

At its heart, this is an excellent guide for HTML novices. It teaches you all you need to know about the language as well as placing it within the context of building useable Websites. It also takes a quick look at the ancillary technologies, such as CGI and Java, and provides practical advice on how you can implement these in your Website. The writing style is accessible and light, which enables you to learn without feeling intimidated.

VERDICT **4**/5



DREAMWEAVER MX H.O.T

Authors **Garo Green** Publisher **Peachpit Press**
Price **£33.99** ISBN **0-321-11271-7**

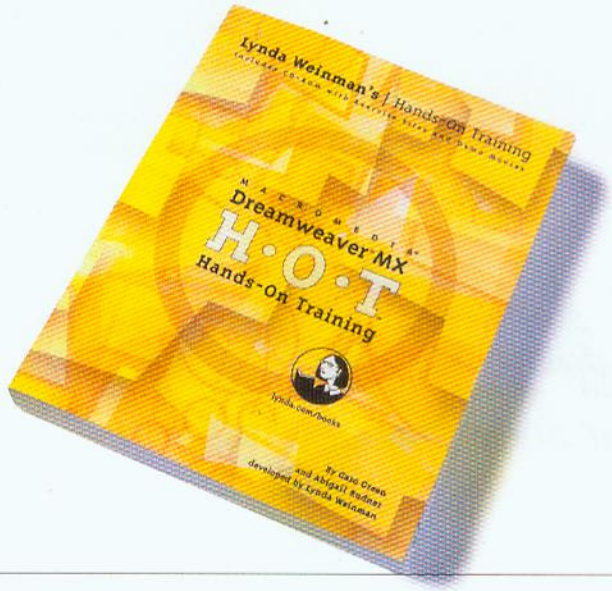
One of the latest in the H.O.T (Hands-on Training) series, this superb overview of Macromedia's professional Web design package is an essential read for anyone who's trying to get to grips with the program for the first time.

The power that *Dreamweaver MX* places into the user's hands is stunning, but the program has been criticised for being hard to learn due to its interface. This book brushes these concerns aside and offers a complete course that uses practical exercises to teach each feature of the program. It's a style that's easily accessible to all.

Divided into 20 chapters, each dissects a different area of *Dreamweaver*, providing masses of technical insight into its inner workings. This attention to detail when discussing particular aspects of the program makes the book useful whatever your operating system – PC or Mac.

With the now familiar guided exercises on the accompanying CD, plus a trial version of *Dreamweaver MX*, this is the perfect opportunity to take the application for a test drive.

VERDICT **5**/5



BUILD YOUR OWN WEBSITE

Authors **David Karlins** Publisher **Osborne**
Price **£17.99** ISBN **0-07-222953-5**

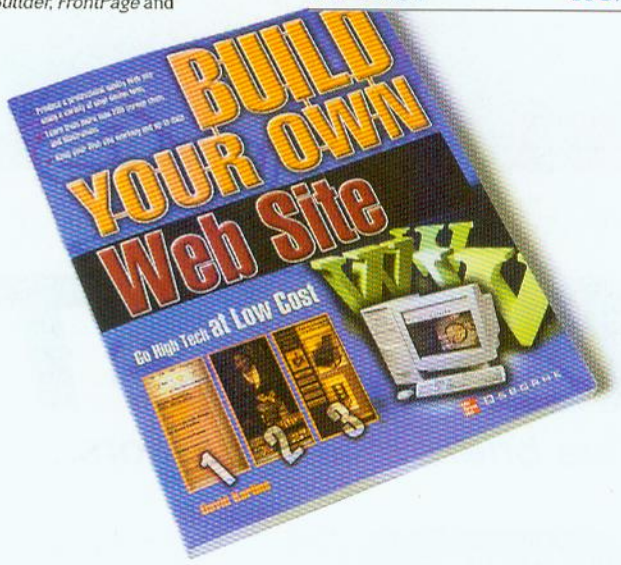
This book was written in response to the needs of readers who either felt that the Web design books they'd read took too much for granted, or left the reader with no idea where to go after they'd turned the last page.

Progress through each of the nine chapters is slow, as you build up your knowledge step by step. A slim volume like this can't hope to be a comprehensive overview of this massive subject; instead, the author has attempted to distil the information you need to build a Website to its most basic components.

The whirlwind tour of the GeoCities *PageBuilder*, *FrontPage* and *Dreamweaver* enables you to see first hand how some of the theory you've been reading about these packages can be put to practical use.

If you're starting out on Website design and have no idea where to begin, this book invites you to step back from the subject and evaluate what information you actually need, before pointing you in the right direction to attain the results you're after. This is an excellent example of a complete beginner's book.

VERDICT **4.5**/5



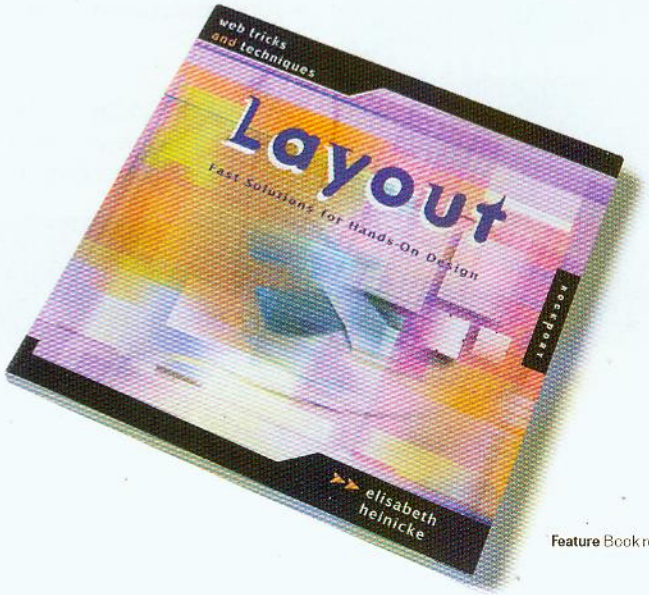
LAYOUT: FAST SOLUTIONS FOR HANDS-ON DESIGN

Author **Elisabeth Heinicke** Publisher **Rockport**
Price **£25.00** ISBN **1-56496-859-6**

In a nutshell, this book offers the intermediate HTML user a self-contained course in more advanced coding. Once you're confident with the basic structure of HTML, you can then increase your knowledge and make your Web pages stand out from the crowd, with text, graphic and layout elements.

Using copious amounts of illustrations, each of the fundamental elements that make up a Website is covered, from text layout to splash pages. Things conclude with a brief introduction to styles. Coding examples litter each page, enabling you to see first hand how each command actually affects your page layout.

There's a glossary, but it omits many of the terms used throughout the book, which can make some of the more technical passages a little confusing – and in some cases impenetrable – to all but design professionals.

That said, this offers more than an intermediate HTML course. Its design makes highly technical layout components easy to understand, because you can see on the page how each tag behaves. The techniques presented here are challenging, but you'll be rewarded with a more engaging Website.

VERDICT **3.5**/5